

HYDROLIGHT 4.0 USERS' GUIDE

Curtis D. Mobley
Sequoia Scientific, Inc.

19981204 024

SEQUOIA

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

PROJECT 33 FINAL REPORT

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE

November 1998

3. REPORT TYPE AND DATE COVERED

Final Report

4. TITLE AND SUBTITLE

HYDROLIGHT 4.0 USERS' GUIDE

5. FUNDING NUMBERS

N00014-98-C0001

6. AUTHOR(S)

Curtis D. Mobley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Sequoia Scientific, Inc.
9725 S.E. 36th St. Suite 308
Mercer Island, WA 98040

8. PERFORMING ORGANIZATION REPORT NUMBER

Project 33 Final Report

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Environmental Optics Program Code 3220P
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000
Attn: Dr. Joan Cleveland

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 Words)

HYDROLIGHT 4.0 is a radiative transfer numerical model that computes spectral radiance distributions and related quantities for water bodies. Input to the model consists of user-supplied information about the absorbing and scattering properties of the water body, the sky radiance incident onto the water surface, the wind speed, and the bottom of the water column. The water absorbing and scattering properties can vary with depth and wavelength. The model includes internal sources and inelastic scattering. Output from the model includes the radiance distribution within and leaving the water body as a function of depth, direction, and wavelength as well as various irradiances, reflectances, and diffuse attenuation functions. This report is a self-contained guide to running the HYDROLIGHT 4.0 model, which includes a Graphical User Interface for personal computers with Microsoft Windows 95/98/NT, Microsoft Excel spreadsheet analysis of output, automatic file management, and easy input of users' data.

14. SUBJECT TERMS

Hydrolight, optical oceanography, radiative transfer, ocean color, remote sensing

15. NUMBER OF PAGES

96

16. PRICE CODE

17. SECURITY CLASSIFICATION
OF REPORT

UNCLASSIFIED

18. SECURITY CLASSIFICATION
OF THIS PAGE

UNCLASSIFIED

19. SECURITY CLASSIFICATION
OF ABSTRACT

UNCLASSIFIED

20. LIMITATION OF ABSTRACT

Unlimited

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Philosophical Comments, Warnings, and Excuses	3
2. OVERVIEW OF HYDROLIGHT 4.0	5
2.1 Ways in Which HYDROLIGHT Can Be Used	5
2.2 The Physical Model	6
2.3 The Mathematical Model	8
2.4 Input	10
2.5 Output	12
3. HYDROLIGHT ORGANIZATION	14
3.1 The HYDROLIGHT 4.0 Software Package	14
3.2 File Naming Conventions	16
3.3 Directory Structure	18
4. STANDARD RUNS	22
4.1 Example of Using the Graphical User Interface	24
4.2 The Default Printout	33
4.3 Examining Output with Excel	36
4.4 Plotting Output with IDL	40
4.5 Setting Defaults	40
5. SPECIAL RUNS	42
5.1 Discretizing a Phase Function	42
5.2 Creating a New Surface File	45
6. CREATING INPUT ROUTINES AND DATA FILES	49
6.1 Subroutines for IOP Models	49
6.2 Subroutines for Chlorophyll Distributions	53
6.3 Subroutines for CDOM Absorption	53
6.4 Subroutines for Sky Models	54
6.5 Subroutines for Bioluminescence	55
6.6 Data Files for WETLabs ac-9 Data	55
6.7 Data Files for Bottom Reflectances	57
7. ADVICE ON USING HYDROLIGHT 4.0	59
7.1 Depth Considerations	59
7.2 Wavelength Considerations	61
7.3 Inelastic Scattering and Bioluminescence	62

8. EXAMPLE HYDROLIGHT RUNS	64
8.1 Discretizing a Phase Function	64
8.2 Creating a Surface Wind File	64
8.3 A Simple Radiative Transfer Simulation	65
8.4 A Simulation of Case 2 Water	67
9. REFERENCES	75
APPENDIX A. DETAILED DESCRIPTION OF THE RUN-TIME INPUT FOR STANDARD RUNS	77
APPENDIX B. INSTALLING THE HYDROLIGHT CODE	84
B.1 Computer Requirements	84
B.2 Installation on Microsoft Windows Operating Systems	85
B.3 Installation on UNIX Operating Systems	89
B.4 The Lahey Automake Configuration File Editor	91
APPENDIX C. LICENSE AGREEMENT FOR USE OF THE LAHEY FORTRAN COMPILER	92
APPENDIX D. LICENSE AGREEMENT FOR USE OF HYDROLIGHT 4.0	94

1. INTRODUCTION

General knowledge of the radiance distribution within and leaving a water body is a prerequisite for the solution of many problems in underwater visibility, "ocean color" remote sensing, biological primary productivity, and mixed-layer thermodynamics. Moreover, because radiance is the fundamental radiometric quantity, all other quantities of interest to optical oceanographers—various irradiances, diffuse attenuation functions, reflectances and the like—can be computed from their definitions once the radiance is known.

HYDROLIGHT 4.0 is a radiative transfer numerical model that computes radiance distributions and derived quantities for natural water bodies. In brief, this model computes from first principles the time-independent radiance distribution within and leaving any plane-parallel water body. Input to the model consists of the absorbing and scattering properties of the water body, the nature of the wind-blown sea surface and of the bottom of the water column, and the sun and sky radiance incident on the sea surface. Output consists of archival printout and of files of digital data, from which numerical, graphical or spreadsheet analyses can be performed.

The model is designed to solve a wide range of problems in optical oceanography and limnology. The input absorbing and scattering properties of the water body can vary arbitrarily with depth and wavelength. These inherent optical properties (IOPs) can be obtained from actual measurements or from analytical models. Analytical models of the IOPs can build up the total IOPs from contributions by any number of individual components. The input sky radiance distribution can be completely arbitrary in the directional and wavelength distribution of the direct solar and diffuse sky light. In its most general solution mode, HYDROLIGHT 4.0 includes the effects of inelastic scatter by chlorophyll fluorescence, by colored dissolved organic matter (CDOM) fluorescence, and by Raman scattering by the water itself. The model also can simulate internal layers of bioluminescing microorganisms.

This Users' Guide for Version 4.0 of the HYDROLIGHT model assumes that the reader is familiar with the basic terminology and notation of optical oceanography. If this is not the case, then the reader should first consult the review paper by Mobley (1995) or one of the books by Kirk (1994); Spinrad, Carder, and Perry (1994); or Mobley (1994). The Users' Guide gives a general overview of the capabilities of HYDROLIGHT 4.0, describes in detail how to run the model, and shows example output. The Users' Guide is independent of any other publication and should be adequate for users who wish to run HYDROLIGHT as a "black box" model or with minor modifications, such as adding routines to read in the user's data on a special format or adding additional output of interest to the user. The text *Light and*

Water: Radiative Transfer in Natural Waters (Mobley, 1994) describes in considerable detail the mathematical methods employed in HYDROLIGHT 4.0. That book is the primary technical documentation for the HYDROLIGHT 4.0 model. The source code itself is documented by references to the equations of *Light and Water*.

The HYDROLIGHT source code is written entirely in FORTRAN 77, in order to make it easily portable to almost any computer with a FORTRAN 77 or FORTRAN 90 compiler¹. All input and output files are written as ASCII files, to assure easy transfer of files between computers with different operating systems. In order to simplify running HYDROLIGHT, two “front-end” programs are provided for running HYDROLIGHT in its standard mode (the usual way of running HYDROLIGHT, which will account for the vast majority of runs made by most users). One front end is a Graphical User Interface (GUI) that runs *only* on computers running the Microsoft Windows 95®, 98®, or NT® operating systems. The other front end is a text-based “question-and-answer” program written in FORTRAN 77; this front-end program is provided for users with other operating systems. The GUI and text-based front end programs are provided only to make HYDROLIGHT more user friendly; it is not actually necessary to run either front-end program in order to run HYDROLIGHT itself. As with previous versions, users can create or modify the needed input files with a text editor and then submit HYDROLIGHT runs from a command window (e.g., a DOS window on a PC or a Terminal window on a UNIX workstation). This independence of HYDROLIGHT from the front-end programs makes it easy to couple HYDROLIGHT with other models, as for example in coupled biological-optical-physical ecosystem models.

Throughout this report, the names of mathematical variables are written in italics, e.g., *U*, *z*, or *zeta*. The names of computer programs, directories, and files are written in a sans serif font, e.g., `abcase1.f` or `Pupcast2.txt`. Path names are written using the DOS format with a backstroke, e.g., `..\data\phasefun\avgpart.dpf`. The same path on a UNIX machine would be written with a slash, e.g., `../data/phasefun/avgpart.dpf`. User input to and output from programs is shown in Courier. Options on Graphical User Interfaces are shown in SMALL CAPS.

1. Although it is popular to deride FORTRAN as an ancient language no longer spoken in computer science departments, the fact remains that FORTRAN is, and likely will remain, the best language for doing numerical computations such as solving differential equations—which is what HYDROLIGHT does for 90% of its run time. I have done comparison calculations with one of the currently popular languages and found it to be as much as 100 times slower than FORTRAN. I make no apology for keeping HYDROLIGHT in FORTRAN.

1.1 Philosophical Comments, Warnings, and Excuses

It is important to understand that the HYDROLIGHT model per se is a radiative transfer model, not a model of oceanic optical properties. You, the user, must supply the inherent optical properties and boundary conditions to the HYDROLIGHT core code. (Indeed, the HYDROLIGHT model is not even restricted to the oceanic setting, although that is our interest here. If you supply HYDROLIGHT with the optical properties of orange paint, for example, HYDROLIGHT will happily solve for the radiance distribution within and leaving the paint.) This requirement for user input gives HYDROLIGHT great power—each user can run the model for exactly the conditions of interest—but also requires considerable forethought and effort on the user's part, in order to specify IOPs and boundary conditions that correctly describe the water body of interest. Some users of previous versions have not understood their role in providing input to HYDROLIGHT, so let me say this again, with more emphasis:

HYDROLIGHT does not know the inherent optical properties, or the chlorophyll profile, or the depth, or anything else about the water body *you* are interested in. *You* must provide this information to HYDROLIGHT. The various IOP models, phase functions, chlorophyll data sets, ac-9 data sets, etc. that come with HYDROLIGHT are *examples* of how to provide IOP and other information to HYDROLIGHT. *You* will need to replace these example routines and data sets with your own, in order to simulate the water body of interest to you.

As with any model, HYDROLIGHT's output is only as good as the input provided by the user. If you are using a simple "case 1" water model for the absorption and scattering coefficients, guessing the phase function, and using the mid-ocean atmospheric parameters in the sky radiance model, you should not be surprised (and are not allowed to complain) if the HYDROLIGHT-predicted water-leaving radiances differ significantly from what you measured in your experiment in case 2 coastal waters. Likewise, if you give HYDROLIGHT a file with thousands of ac-9 data points generated by sampling at 6 Hz during a depth profile, HYDROLIGHT will do its best to solve the radiative transfer equation with *exactly* the absorption and scattering profile you have given it. The result may be numerically disastrous if the ac-9 profile looks like random instrumental noise superimposed on a smoother signal, as illustrated in Fig. 1.

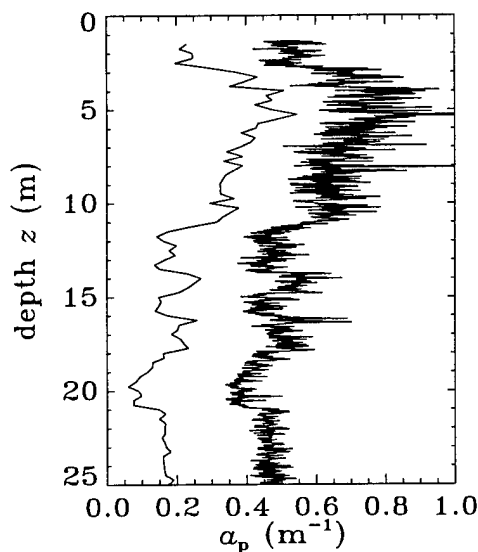


Figure 1. Smoothed and unsmoothed data. The right curve is 2,056 values of particulate absorption a_p obtained from an ac-9; this curve is too noisy to be used as input to HYDROLIGHT. The left curve (offset to the left by 0.3 for clarity) is the same data binned into 25 cm depth bins to give 95 a_p values; this curve is suitable for input to HYDROLIGHT and still contains adequate depth resolution of the absorption fine structure.

It is your job to clean up noisy data, smoothing and massaging as much as your conscience and scientific expertise allow, before using it as input to HYDROLIGHT. I have tried to make HYDROLIGHT reasonably robust in checking for bad input, but there is still great opportunity for entering unphysical input and getting unphysical output.

HYDROLIGHT has been a work in progress since I first started working out the numerical algorithms as a postdoc in 1979, and it will continue to be so. The invariant imbedding algorithms at the core of HYDROLIGHT are mature and well debugged after many years of use. However, features such as the Graphical User Interface (GUI) and spreadsheet analysis of output are new with version 4.0 and are likely to evolve quickly as I receive feedback from the initial users of version 4.0.

Version 4.0 was created in response to users' requests for a more user-friendly code and for a code that would run on inexpensive personal computers (as opposed to mainframes or UNIX workstations). Rather than just develop a PC version of HYDROLIGHT version 3, I decided to make major modifications of the code structure and to develop a greatly improved version of HYDROLIGHT. This process took about ten times longer than I anticipated. I still have many ideas for improvements, but time and development funds did not permit their inclusion in release 4.0. Revenues permitting, HYDROLIGHT will continue to improve. Your suggestions for future improvements are most welcome.

2. OVERVIEW OF HYDROLIGHT 4.0

This section gives brief descriptions of how HYDROLIGHT can be used, of what assumptions are built into the code, of what input is required, and of what output can be obtained from a HYDROLIGHT run.

2.1 Ways in Which HYDROLIGHT Can Be Used

Previous versions of HYDROLIGHT have been used in a variety of studies ranging from bio-optical oceanography to remote sensing. Some of the ways in which HYDROLIGHT can be used are as follows:

- HYDROLIGHT can be run with modeled input values to generate in-water light fields, which in turn become the input to models of primary productivity or mixed-layer dynamics. Such information is fundamental to the coupling of physical, biological, and optical feedback models.
- HYDROLIGHT can be run with the IOPs of different water types to simulate in-water light fields for the purpose of selecting or designing instruments for use in various water types. Such information can aid in the planning of field experiments.
- HYDROLIGHT can be run with assumed water inherent optical properties as input, in order to obtain estimates of the signals that would be received by various types or configurations of remote sensors, when flown over different water bodies and under different environmental conditions. Such information can guide the planning of specific operations.
- HYDROLIGHT can be used to isolate and remove unwanted contributions to remotely sensed signatures. Consider the common remote-sensing problem of extracting information about a water body from a downward-looking imaging spectrometer. The detected radiance contains both the water-leaving radiance (the signal, which contains information about the water body itself) and sky radiance reflected upward by the sea surface (the noise). HYDROLIGHT separately computes each of these contributions to the radiance heading upward from the sea surface and thus provides the information necessary to correct the detected signature for surface reflection effects.

- When analyzing experimental data, HYDROLIGHT can be run repeatedly with different water optical properties and boundary conditions to see how particular features of the data are related to various physical processes or features in the water body (such as substance concentrations or external environmental conditions). Such simulations can be valuable in formulating hypotheses about the causes of various features in the data.
- HYDROLIGHT can be used to simulate optical signatures for the purpose of evaluating proposed remote-sensing algorithms for their applicability to different environments or for examining the sensitivity of algorithms to simulated noise in the signature.
- HYDROLIGHT can be used to characterize the background environment in an image. When attempting to extract information about an object in the scene, all of the radiance from the natural environment may be considered noise, with the radiance from the object being the signal. The model can then be used to compute and remove the environmental contribution to the image.
- HYDROLIGHT can be run with historical (climatological) or modeled input data to provide estimates about the marine optical environment during times when remotely or in-situ sensed data are not available.

Such information can be provided in many forms: water-leaving radiances for remote-sensing applications, in-water apparent optical properties (such as K functions) for Lidar bathymetry applications, or ambient light field data as may be relevant to underwater visibility applications.

2.2 The Physical Model

Many problems of interest in optical oceanography and remote sensing can be solved using time-independent radiative transfer theory applied to plane-parallel geometries. The consideration of time-dependent, plane-parallel problems is not as restrictive as it might seem on first glance. For example, although the oceans are horizontally inhomogeneous, the horizontal scales of significant optical variability (typically tens of meters to kilometers) are usually much greater than the vertical scales (tens of centimeters to tens of meters). In this

case we can think of the ocean as consisting of optically independent "patches" of water, for which each patch can be well modeled as a horizontally homogeneous water body whose optical properties vary only with depth. (This is a one-dimensional geometry, with the one dimension being depth). We can then independently apply a one-dimensional radiative transfer model at the center of each patch in order to simulate the entire, horizontally inhomogeneous water body. In the analysis of imaging spectrometer data, one might even apply such a model to the water patch associated with each pixel in the image.

Such a piecewise simulation is justified so long as the horizontal size of each water patch is at least several photon mean free paths. This is usually the case. In the open ocean, photon mean free paths (the inverse of the beam attenuation coefficient) are never more than 50 m (at blue wavelengths, and much less at other wavelengths) in even the clearest waters; horizontal variability in such waters is often on scales of kilometers. In coastal waters subject to river runoff, sediment resuspension, and variable shallow bottom topography, optical properties and boundary conditions can change horizontally on scales of meters to tens of meters. However, such waters tend to be rather turbid and have photon mean free paths of tens of centimeters to a few meters. In either case, the use of a one-dimensional radiative transfer model is justified. The use of time-independent radiative transfer is valid whenever the time scales for changes in environmental conditions (typically seconds to seasons) are much greater than the time required for the light field to assume a steady state within the water body after a change in the optical properties or boundary conditions (milliseconds). Solving a sequence of time-independent, one-dimension radiative transfer problems in order to simulate a changing (in both time and space) water body is computationally much faster than solving one large time-dependent, three-dimensional problem.

Other physical considerations also dictate the generality required in an oceanic radiative transfer model. High absorption by water itself means that little light penetrates the ocean outside of the near-ultraviolet to near-infrared region from 350 to 800 nm. For the purpose of computing energy transfer through the air-water surface, it is often sufficient to account for capillary waves on the sea surface while neglecting the larger gravity waves. Multiple scattering is almost always important, but polarization may be neglected for many applications. Inelastic scattering processes such as Raman scatter by the water itself and fluorescence by chlorophyll and CDOM can in some circumstances make significant contributions to the light field.

The HYDROLIGHT physical model, which addresses the above considerations, can be summarized as follows:

- time-independent
- horizontally homogeneous IOPs and boundary conditions
- arbitrary depth dependence of IOPs
- wavelengths between 350 and 800 nm
- capillary-wave air-water surface
- various bottom options
- includes all orders of multiple scattering
- includes Raman scatter by water
- includes fluorescence by chlorophyll and CDOM
- includes internal sources such as bioluminescence
- does not include polarization
- does not include gravity waves or whitecaps

2.3 The Mathematical Model

The fundamental quantity that describes the time-independent, one-dimensional light field in the ocean is the spectral radiance $L(z, \theta, \phi, \lambda)$, with units of $\text{W m}^{-2} \text{sr}^{-1} \text{nm}^{-1}$. The spectral radiance completely determines the depth (z), directional (θ, ϕ), and wavelength (λ) behavior of the light field. Therefore, all other quantities of interest, such as various irradiances, diffuse attenuation functions (K -functions), and reflectances, can be computed from their definitions once the spectral radiance is known. In order to predict the spectral radiance, HYDROLIGHT solves the integro-differential radiative transfer equation (RTE) along with its boundary conditions. Because of their mathematical complexity, these equations must be solved numerically for any realistic situation (see, for example, Chapter 8 of *Light and Water*).

In HYDROLIGHT the depth z is measured as positive downward from zero at the mean sea surface. The polar angle θ is measured from zero in the nadir (downward) direction, and the azimuthal angle ϕ is measured from zero in the downwind direction. As is conventional in radiative transfer theory, θ and ϕ always refer to the direction of photon travel. The viewing direction—the direction a sensor would point in order to detect the radiance $L(z, \theta, \phi, \lambda)$ —is then $(\theta_v, \phi_v) = (180^\circ - \theta, \phi + 180^\circ)$. Thus, in the printout, a radiance labeled with $\theta = 180^\circ$ refers to radiance heading upward, which is detected by an instrument pointed downward ($\theta_v = 0$).

Any radiance sensor actually measures an average of $L(z, \theta, \phi, \lambda)$ taken over some finite solid angle $\Delta\Omega$, which is determined by the field of view of the instrument, and over some finite bandwidth $\Delta\lambda$, which is determined by the wavelength response of the instrument.

Likewise, in order to solve the RTE numerically, we discretize it by averaging over direction and wavelength. In the HYDROLIGHT model, this directional averaging is performed by first partitioning the set of all directions (θ, ϕ) , $0 \leq \theta \leq 180^\circ$, $0 \leq \phi < 360^\circ$, into regions bounded by lines of constant θ and constant ϕ , plus two polar caps. These quadrilateral regions and polar caps are collectively called "quads." The individual quads Q_{uv} are labeled by discrete indices $u = 1, 2, \dots, M$ and $v = 1, 2, \dots, N$ to show their θ and ϕ positions, respectively. The standard (default) quad layout is shown in Figure 2. In this layout, which has $M = 20$ and $N = 24$, the polar caps have a 5° half angle and the θ boundaries lie at $5, 15, 25, \dots, 75, 85, 90, 95, 105, \dots, 175$ degrees. For mathematical reasons there is no quad centered on the "equator" at $\theta = 90^\circ$. However, the radiances computed for the 85° - 90° and 90° - 95° quads can be averaged to give the "horizontal" radiance at a nominal angle of $\theta = 90^\circ$. Thus the standard quad layout essentially gives 10° resolution in θ and 15° in ϕ .

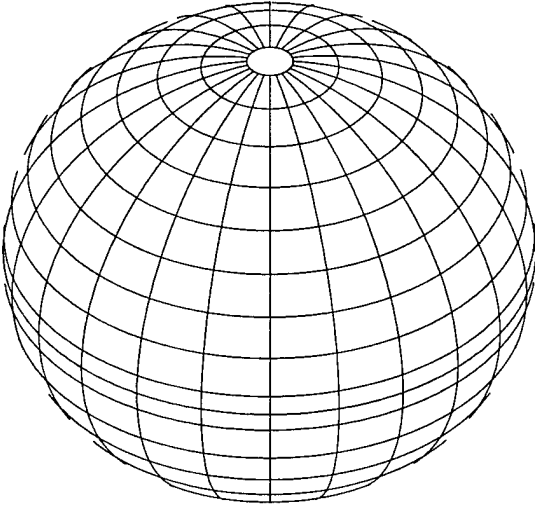


Figure 2. The HYDROLIGHT standard quad layout, which has a nominal angular resolution of $\Delta\theta = 10^\circ$ and $\Delta\phi = 15^\circ$.

Similarly, the wavelength region of interest is partitioned into a number of contiguous wavelength bands of width $\Delta\lambda_j$, $j = 1, 2, \dots, J$. The $\Delta\lambda_j$ need not be the same size for different j values.

The fundamental quantities computed by the HYDROLIGHT 4.0 model are then the quad- and band-averaged radiances at any selected set of depths z_k , $k = 1, 2, \dots, K$:

$$L(k, u, v, j) \equiv \frac{1}{\Delta\Omega_{uv} \Delta\lambda_j} \int_{Q_{uv}} \int_{\Delta\lambda_j} L(z_k, \theta, \phi, \lambda) \sin\theta \, d\theta \, d\phi \, d\lambda.$$

In addition to the radiances within the water, HYDROLIGHT computes the upwelling radiance in all directions (all quads) just *above* the sea surface. This upwelling radiance includes both the water-leaving radiance and that part of the incident direct and diffuse sky radiance that is reflected upward by the wind-blown sea surface. The water-leaving and reflected-sky radiances are computed separately in order to isolate the water-leaving radiance, which is the quantity of interest in many remote sensing applications. The development of the quad- and band-averaged versions of the RTE and of the associated boundary conditions is given in full in *Light and Water*.

It must be noted that the quads "homogenize" the radiance within each quad, much like a frosted-glass window does. Thus, in the quad layout of Fig. 2, it is not possible to resolve the difference in the radiance for polar angles $\theta = 26^\circ$ and $\theta = 34^\circ$, because they both lie in the same quad extending from $\theta = 25^\circ$ and $\theta = 35^\circ$. However, there is a difference in $\theta = 34^\circ$ and $\theta = 36^\circ$, because those angles lie in different quads and thus are represented by different quad-averaged radiances. If it is necessary to have greater angular resolution in the radiance distribution, a different quad layout can be created by the user as described in Section 5.2 below. Note, however, that the computer storage and run time are proportional to M^2N^2 , so increasing the angular resolution comes with a considerable computational cost.

2.4 Input

In order to run HYDROLIGHT to predict the spectral radiance distribution within and leaving a particular body of water during particular environmental (sky and surface wave) conditions, the user supplies the core model with the following information (via direct input or via user-written subroutines or user-supplied data files):

- **The inherent optical properties of the water body.** These optical properties are the absorption and scattering coefficients and the scattering phase function (which are equivalent to the volume scattering function, the beam attenuation coefficient, and the albedo of single scattering). These properties must be specified as functions of depth and wavelength.
- **The state of the wind-blown sea surface.** Version 4.0 of HYDROLIGHT models the sea surface using the Cox-Munk capillary wave slope statistics, which adequately describe the optical reflection and transmission properties of the sea surface for moderate wind speeds and solar angles away from the horizon. In this case, only the

wind speed needs to be specified.

- **The sky spectral radiance distribution.** This radiance distribution (including background sky, clouds, and the sun) can be obtained from semi-empirical models that are built into HYDROLIGHT, from observation, or from a separate user-supplied atmospheric radiative transfer model (such as LOWTRAN).
- **The nature of the bottom boundary.** The bottom boundary is described in terms of a bi-directional radiance reflectance function (BRRF). For finite-depth bottoms, the BRRF is computed from the given irradiance reflectance of the bottom. For infinitely deep water, the inherent optical properties of the water body below the region of interest are used to compute the needed BRRF.

The absorption and scattering properties of the water body can be provided to the HYDROLIGHT model in various ways. For example, if actual measurements of total absorption are available at selected depths z and wavelengths λ , then these values can be read from a file provided at run time. An interpolation scheme can be used to define absorption values for those z and λ values not contained in the data set. In the absence of actual measurements, the total absorption of the water body can be modeled in terms of contributions by any number of components. Thus the total absorption can be built up as the absorption by water itself, plus the absorption by chlorophyll-bearing microbial particles, plus that by CDOM, by detritus, by mineral particles, and so on. In order to specify the absorption by chlorophyll-bearing particles, for example, you can specify the chlorophyll profile of the water column and then use a bio-optical model to convert the chlorophyll concentration to the needed absorption coefficient. The chlorophyll profile also provides information needed for the computation of chlorophyll fluorescence effects. Each individual absorption component has its own depth and wavelength dependence. Similar modeling can be used for scattering.

Phase function information is often provided by using a Rayleigh-like phase function for scattering by the water itself, by using a Petzold-like phase function for scattering by particles, and by assuming that dissolved substances like CDOM do not scatter. The individual-component phase functions are weighted by the respective scattering coefficients and summed in order to obtain the total phase function.

HYDROLIGHT also requires the downwelling radiance incident onto the sea surface as input. The HYDROLIGHT model does not carry out any radiative transfer calculations for the atmosphere *per se*. However, the sky radiance for either cloud-free or overcast skies can

be estimated from simple analytical models or from semi-empirical models; such models are provided as a part of the HYDROLIGHT code. Alternatively, if the sky radiance is actually measured, that data can be used as input to HYDROLIGHT via a user-written subroutine. It is also possible to run an independent atmospheric radiative transfer model such as LOWTRAN (Kneizys, et al., 1988) in order to generate the sky radiance coming from each quad of the sky hemisphere, and then give the LOWTRAN-generated values to HYDROLIGHT as input.

For finite-depth water columns, the bottom boundary BRRF is computed from the specified irradiance reflectance of the bottom and the assumption that the bottom is a Lambertian reflector. In general, the bottom reflectance is a function of wavelength and depends on the type of bottom—gray mud, coral sand, green algae, etc. For infinitely deep water columns, the (non-Lambertian) BRRF is computed using the IOPs at the deepest depth *where output is requested* in the simulation at hand. For a remote sensing simulation concerned only with the water-leaving radiance, it is usually sufficient to solve the radiative transfer equation only for the upper two “diffuse attenuation depths” [a depth of $2/K_d(\lambda)$], because almost all light leaving the water surface comes from this near-surface region. In this case, the bottom boundary condition can be taken to describe an optically infinitely deep layer of water below the depth corresponding to two diffuse attenuation depths. In a biological study of primary productivity, it might be necessary to solve for the radiance down to five (or more) optical depths, in which case the bottom boundary condition would be applied at that depth. In such cases HYDROLIGHT computes the needed bottom boundary information from the inherent optical properties at the deepest depth of interest.

2.5 Output

Output from HYDROLIGHT consists of both “printout” (an ASCII file formatted for hardcopy printing) and files of digital data. The default printout gives a moderate amount of information to document the input to the run and to show the quantities of interest to most oceanographers (such as various irradiances, reflectances, mean cosines, irradiance K -functions, and zenith and nadir radiances). The printout is useful for taking a quick look at the results of a run, or for cutting and pasting a particular part of the output into another document or spreadsheet. Optionally, the printout can give the full radiance distribution (separated into direct and diffuse components), radiance K -functions, elastic-scatter path functions, and the like. The printout is easily tailored to the user’s requirements. Section 4.2 describes the printout in more detail.

A file of digital data contains the complete output from the run, including the full radiance distribution as a function of depth, direction, and wavelength. This file is generally used as input to plotting routines to obtain graphical output of various quantities as functions of depth, direction, or wavelength. Routines for graphical output are not a part of the HYDROLIGHT code because of the wide variety of graphics packages in use and because different users generally want different kinds of plots. However, a few plotting routines written in the IDL language are included with HYDROLIGHT as a convenience for users who have that popular software package. The digital output file is formatted to facilitate the opening of the file by IDL (which stores arrays by rows, rather than by columns as in FORTRAN). Section 4.4 comments further on the use of IDL in plotting HYDROLIGHT output.

Two other files of digital output are formatted for use within Microsoft Excel® spreadsheets. One file is formatted to facilitate the analysis of data one wavelength at a time. For example, you might want to plot various irradiances as a function of depth at one wavelength. The other file is formatted to facilitate the analysis of one variable at a time, as a function of depth and wavelength. For example, you might want to plot the absorption coefficient as a function of depth and wavelength. Excel macros are provided to open these files within Excel® and automatically generate spreadsheets containing the quantities of interest to most oceanographers. Section 4.3 describes how to convert the HYDROLIGHT output files into Excel spreadsheets.

3. HYDROLIGHT ORGANIZATION

This section describes how the HYDROLIGHT code is organized. The different types of routines and data files are described, and the file-naming conventions and directory structure are established.

3.1 The HYDROLIGHT 4.0 Software Package

The HYDROLIGHT 4.0 package consists of many different main programs, subroutines, and data files. The different routines and files can be classified as follows:

- **Core FORTRAN main programs and subroutines.** Each of these routines is flagged in the source code by a statement of the form "Core routine on file filename.f" where "filename" is the name of the file containing the routine. These core routines include both HYDROLIGHT-specific routines and various public-domain subroutines taken from sources such as LAPACK (Linear Algebra PACKage) and BLAS (Basic Linear Algebra Subroutines; see Dongarra and Grosse, 1987). The public-code routines are used by the HYDROLIGHT-specific routines for mathematical tasks such as matrix inversion, eigenvector-eigenvalue analysis, and solving differential equations. Only the most sophisticated users would even contemplate tampering with any of these highly mathematical core routines.
- **Templates for creating user-supplied subroutines and data files.** Each of these routines is flagged in the source code by a statement of the form "Template for user-supplied ... routines." The ellipsis describes the type of routine, e.g. "sky radiance" or "absorption and scattering." These templates show the required formats for various routines that provide the core program with information about the absorbing and scattering properties of the particular water body being simulated, about the sky radiance distribution, and the like. Many users will want to write their own versions of such routines in order, for example, to read in their own measured absorption and scattering profiles or to insert their own analytical models of the inherent optical properties. This can be done by inserting the desired code into the corresponding template.

- **Example user-supplied subroutines.** Each of these routines is flagged in the source code by a statement of the form “Example user-supplied ... routine on file filename.f” The ellipsis describes the type of routine, e.g. “sky radiance” or “absorption and scattering.” Examples of these standardized subroutines are provided with the HYDROLIGHT code. These examples illustrate how the template routines can be expanded to provide routines for IOP input, sky input, etc. The example user routines can be used as is, or can be modified as desired to alter the input to HYDROLIGHT.
- **Data files.** These files provide input such as scattering phase functions, sea-surface reflectance and transmission properties for different wind speeds, bottom reflectances, and atmospheric properties used by the default sky irradiance models. The distributed files will be sufficient for many users. However, users can add to this collection by making “specialized” runs (for example, to add additional phase functions) as described in Section 5.
- **Plotting routines written in IDL** (Interactive Data Language; IDL® is a product of Research Systems, Inc.). These routines are, strictly speaking, not a part of the HYDROLIGHT code. A small collection of IDL routines is included for the convenience of users who have the IDL software package. Users may wish to discard these routines and use other software for graphical analysis of the HYDROLIGHT digital output.
- **Example simulations.** The input, run script, and output files for a few typical Hydrolight simulations are given for reference. The run scripts are given for both DOS and UNIX systems. Users should reproduce these simulations on their own computers after installation, in order to verify that HYDROLIGHT is running properly on their computer.

The FORTRAN routines of the HYDROLIGHT 4.0 code are grouped into three parts found on directories `maincode`, `surfcode`, and `discpf` (see Section 3.3). These routines carry out both “standard” and “special” runs.

Standard runs are those that result in a solution of the radiative transfer equation. The routines for performing standard runs are found in the `maincode` directory. The `maincode` routines use the available collection of phase-function, sea-surface, and other data files along with the user input provided at run time to solve the radiative transfer problem defined by the

input. Standard runs are often made in a series, with only minor changes in the input for each run (e.g., a change in the solar zenith angle or in the bottom depth, with all other input being held constant from one run to the next). Standard runs will constitute almost all of the HYDROLIGHT runs made by most users.

Special runs are those that are made for adding a new phase function or a new wind speed to the available collection of data files. Special runs are made only once for a given phase function or wind speed. The routines in directory `surfcode` perform the calculations associated with the air-water surface boundary. These surface calculations depend only on the wind speed and on the quad layout; they are independent of the sky radiance distribution and of the water inherent optical properties. Therefore, the surface calculations need to be performed only once for a given wind speed and quad layout. The results are saved (in directory `data\surfaces` as the files named `surfwind.U`, where `U` is the wind speed) for repeated use by the main code, which performs the remainder of the computations. Only users who need additional wind speeds or who want to customize the quad layout will ever need to run the surface routines. The use of the routines in `surfcode` is described in Section 5.2.

The routines in directory `discpf` are used to prepare a scattering phase function for use by the HYDROLIGHT main code. (This preparation is called discretizing the phase function.) These calculations need to be done only once for a given phase function and quad partition. The use of the routines in `discpf` is described in Section 5.1.

3.2 File Naming Conventions

For the convenience of users running HYDROLIGHT on PCs with Windows 95, which at its core retains the DOS “8.3” file naming convention, file names in this users’ guide and in the distributed code follow the DOS convention of eight-character file names with three-character extensions, e.g. `filename.f` for FORTRAN source code, `filename.txt` for ASCII text files, or `filename.bat` for script (batch) files. Users with other operating systems can feel free to use longer file names when running HYDROLIGHT.

Part of the input to a HYDROLIGHT run is a single-word “root” name that is used to construct the names of all input and output files associated with that run. Thus, after running the front-end program, the input to HYDROLIGHT will be found on a file named `lroot.txt`, where “root” has been replaced by the chosen identifying name, e.g. `lupcast2.txt` if `root = upcast2`. After the run finishes, the printout will be found on file `Proot.txt` (e.g., `Pupcast2.txt`), and so on. The following set of files is created for every HYDROLIGHT run:

Droot.txt	the file containing the full <u>D</u> igital output from the run (usually used as input to a graphics package; see Section 4.4). Written by HYDROLIGHT to the digital directory.
Iroot.txt	the file containing the <u>I</u> nput specifying the run (see Appendix A for a detailed description of this input). Written by the front-end program to the run directory.
Mroot.txt	the file containing the <u>M</u> ulti-wavelength-format output for postprocessing with Excel (see Section 4.3). Written by HYDROLIGHT to the excel directory.
Proot.txt	the file containing the <u>P</u> rintout from the run (usually used for a “quick look” at the output; see Section 4.2). Written by HYDROLIGHT to the printout directory.
Rroot.bat	the file containing the <u>R</u> un script for the run. This file will be slightly different for DOS and UNIX machines. Written by the front-end program to the run directory.
Sroot.txt	the file containing the <u>S</u> ingle-wavelength-format output for postprocessing with Excel (see Section 4.3). Written by HYDROLIGHT to the excel directory.

The above files are of direct interest to the user because they contain the input and output for a HYDROLIGHT run. There are additional files associated with each run, but whose existence is not of interest to most users. Nevertheless, it should be noted that the names of some subroutines needed to run the core code are provided via “include” files, which have names of the form *.inc. These files are all created automatically by the front-end program. For example, if you select the IOP model named “abcase1” (see Section 4.1), then an include file named abscat.inc will contain the call to subroutine abcase1

3.3 Directory Structure

The HYDROLIGHT package is installed by default on the user's computer as a main directory named H40. On a PC, this directory might have the full path name c:\H40; on a UNIX machine the full path might be /usr/home/H40. You can, if desired, rename the main directory (for example, if you wish to have more than one copy of HYDROLIGHT installed for work on different projects, or if different users are working on the same computer and each has his or her own copy of HYDROLIGHT.) In any case, the HYDROLIGHT code uses path

names relative to the main directory (which is referenced as ".." in either DOS, UNIX, or the 95/98/NT operating systems), so that the actual name of the main directory does not matter. The main H40 directory has a number of subdirectories *whose names must not be changed*. All of the computations within a HYDROLIGHT run are done within the subdirectories. Figure 3 shows the layout and names of the subdirectories.

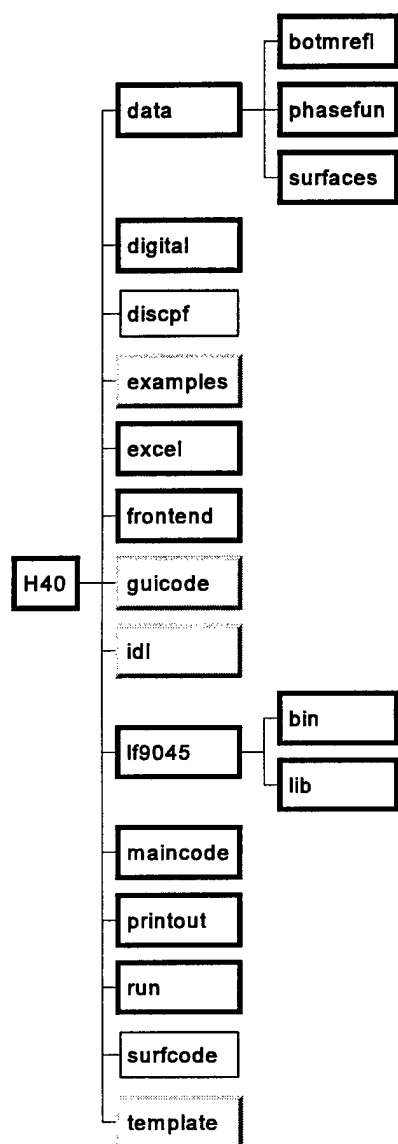


Figure 3. The H40 directory structure. The heavy boxes show the directories involved in HYDROLIGHT standard runs. The light boxes show directories used in special runs. The shaded boxes show directories containing files that may be of interest to some users, but which are not required for HYDROLIGHT runs.

The contents of the various directories are as follows:

- data** contains the input data files for a run. Initially, this directory contains files with the pure water absorption and scattering coefficients and a few example files of ac-9 and chlorophyll data. *Users can add their own data files to this directory, which is the default location where HYDROLIGHT looks for data files.* This directory also has three subdirectories for holding particular types of input:
- botmrefl** contains files of bottom reflectances. Initially, this directory contains a few example files (named *.bot) for different bottom types. Users can add their own files of measured bottom reflectances to this directory.
- phasefun** contains files of discretized phase functions created by the code in directory **discpf**. Initially, this directory contains files (named *.dpf) for various phase functions. Users can add their own files of discretized phase functions to this directory.
- surfaces** contains files of sea-surface information for different wind speeds. These files were created by the code in the **surfcode** directory.
- digital** is the directory where HYDROLIGHT writes the full-output digital data file (the Droot.txt files) from each run.
- discpf** contains the code needed for discretizing a phase function and creating new ..\data\phasefun*.dpf files. See Section 5.1.
- examples** contains the input, run script, and output files for the example simulations.

- excel** contains files associated with Excel spreadsheets. This directory initially contains the two macros (files **singlewl.xls** and **multiwl.xls**) that open the corresponding HYDROLIGHT spreadsheet output files (files **Sroot.txt** and **Mroot.txt**, respectively) as Excel workbooks. HYDROLIGHT writes its spreadsheet output files to this directory.
- frontend** contains the executable GUI front-end program for Windows (file **H40WinFE.exe**) and the source code for the corresponding text-based front-end program (file **H40TxtFE.f**).
- guicode** contains the Visual Basic 5.0 source code for the GUI front end. (Users who do not wish to modify the GUI will not need this code.)
- idl** contains example IDL routines. These routines are designed to read the full-output digital data files. (Users who do not use IDL will not need this code.)
- lf9045** contains the subset of the Lahey FORTRAN 90 version 4.5 compiler that is necessary to compile and link HYDROLIGHT (without modification) on PCs. This directory will not be on your copy of the code if you did not receive the Lahey compiler with HYDROLIGHT. The **bin** and **lib** subdirectories contain the actual Lahey compiler files.
- maincode** contains the FORTRAN source code for the HYDROLIGHT standard runs. (After compilation of the source code, this directory will also contain the corresponding object code and executable file **maincode.exe**).
- printout** is the directory where HYDROLIGHT writes the printout (file **Proot.txt**) from each run.
- run** contains the input (**Iroot.txt**) and run script (**Rroot.bat**) files needed to run the HYDROLIGHT main code. These files are written by the front-end program.

surfcode contains the source code needed to create surface data files (the files named `..\data\surfaces\surfwind.U`, where U is the windspeed in m s^{-1}) for additional wind speeds or different quad layouts. Many users will never need this code.

template contains templates for creating various user-defined subroutines (e.g., for IOP or sky models) and data files (e.g., for ac-9 or chlorophyll profiles).

4. STANDARD RUNS

Standard HYDROLIGHT runs are runs that solve the RTE for a given set of water IOPs and surface and bottom boundary conditions. Because this is the mode of running HYDROLIGHT that is employed most of the time (and all of the time by some users), it is described first. Most users will find the standard quad layout seen in Fig. 2 and the selection of available wind speeds (0, 2, 5, 10, and 15 m s⁻¹) to be adequate. If this is not the case, then special runs to define new wind speeds or quad layouts must be made before making the desired standard runs. These special runs are described in Section 5.

A standard run takes information about the inherent optical properties (IOP's) of the water body and adds information about the sea-surface, the incident sky radiance distribution, and the bottom boundary condition. This information together completely defines a physical situation for which the radiative transfer equation (RTE) has a unique solution. The HYDROLIGHT main code then solves the quad-averaged RTE and generates the final output of the HYDROLIGHT model. The standard computations are entirely analytical; no Monte Carlo simulations are used to solve the RTE within the water body.

There are three ways to initiate a HYDROLIGHT standard run: (1) by running the GUI front-end program (Windows 95/98/NT systems only), (2) by running a text-based front-end program, or (3) by creating the input and run script files with an ASCII text editor and then running HYDROLIGHT from a command window. Most users—especially those who are new to HYDROLIGHT or who are not experienced programmers—prefer to run HYDROLIGHT under control of a front-end program. Experienced programmers sometimes may find option (3) to be the quickest way to submit a new run. However, option (3) requires the user to be familiar with the input format, which is described in detail in Appendix A.

Standard runs are often made in a series in which only one input variable such as the solar zenith angle, water absorption coefficient, or bottom depth is changed between each run. Therefore, the front-end programs have the option of saving the input values from the current run and using those values as the default values for the input to the next run. After making the first run of a series, the user then can quickly “click through” the GUI windows or “hit return” in response to the questions in the text-based front end, pausing only to change the one input value that is to be different in the next run. Similarly, a user who is familiar with the input format can quickly open the input file with a text editor, change a number, re-save the file, and submit the new run.

Regardless of how a run is initiated, the user must give HYDROLIGHT input that defines a unique radiative transfer problem for the environmental conditions of interest. When

running the GUI front end, these specifications are made by selecting allowed values for various parameters, by entering the names of files containing data such as chlorophyll profiles, or by entering a list of depths where the output is to be saved. When running a text-based front end, the specifications are made by responding "yes" or "no" ("y" or "n") to questions, or by typing in file names or lists of values.

The purposes of the front-end programs are to

- guide users through the specification of all of the input needed to define the radiative transfer problem
- use the users' responses to generate the various files needed to run HYDROLIGHT
- run HYDROLIGHT to solve the specified radiative transfer problem

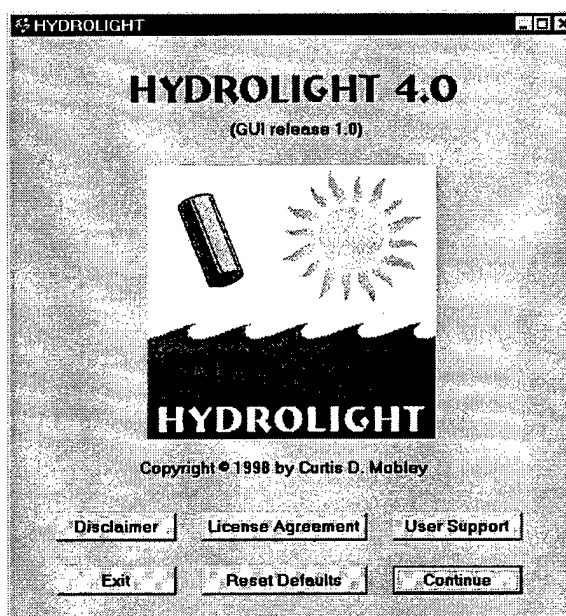
The GUI front-end program consists of a number of windows, each of which prompts the user for a particular type of information, e.g., what IOP model to use, what bottom boundary condition to use, or what depth resolution is desired in the output. The text front-end program obtains the same information by asking a series of questions from a DOS or UNIX command line prompt.

The front-end programs have the option of saving the input, so that the responses from one pass through the front end become the default responses for the next pass. This makes it possible to cycle through HYDROLIGHT runs in a nearly automated fashion in order to carry out a detailed study of how the marine light field depends on some quantity of interest, such as the sun's location or the IOPs of the water body. After the first pass through the front end, only minimal input (such as changing the solar zenith angle or the depth of the bottom) is required on the subsequent passes.

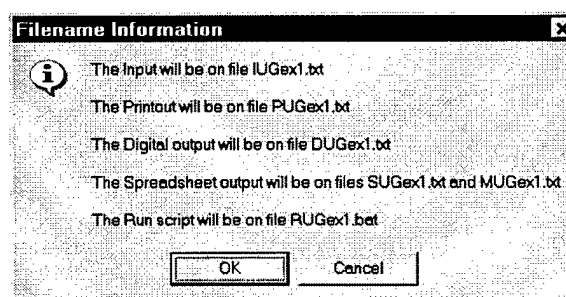
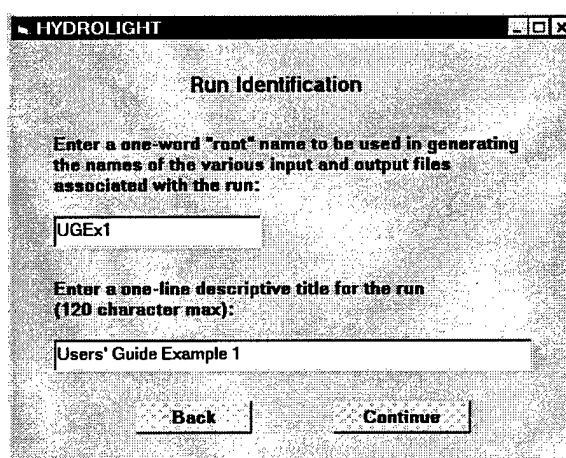
The following pages briefly describe the input requested in the various windows of the GUI front end.

4.1. Example of Using the Graphical User Interface

The Opening form. When running HYDROLIGHT for the first time, click the RESET DEFAULTS button. You will need to read the DISCLAIMER and the LICENSE AGREEMENT before continuing. On subsequent runs, you can proceed immediately by clicking CONTINUE. Any time you wish to return to the default values (as distributed with the code) for the various parameters, just click RESET DEFAULTS.



The RUN IDENTIFICATION form. This form first asks for the "root" name to be used in generating the file names for the run, as described in Section 3.2. For DOS-based machines (e.g., for Windows 95), keep this name to 7 or fewer letters to avoid potential problems with longer names being truncated. For Windows 98/NT, you can use longer names, but keep the root as one word (i.e., do not have any blanks in the name). (When you click CONTINUE, a popup box will show you the names of the files that will be generated.) Also enter a descriptive title for the run. This title will appear on the printout and other output files generated by the run.



The IOP SPECIFICATION form. This form requires you to pick an IOP model, i.e., a model for the absorption and scattering coefficients and scattering phase functions that describe the water body. The input required by various IOP models differs; the needed information will be requested on the next form, depending on which option is selected on this form. Four options are available on this form:

HYDROLIGHT

Inherent Optical Property Specification

Select the "ab" routine that will be used to specify the absorption coefficient (a) and scattering coefficient (b) as functions of depth and wavelength:

☐ ABCONST: a and b are independent of depth (single-wavelength runs only)

☐ ABCASE1: a and b are obtained from bio-optical models for Case 1 water

☐ ABAC9: a and b are obtained from a USER-SUPPLIED file of ac-9 data

☐ OTHER: a and b are obtained from a USER-SUPPLIED model

Back Continue

☐ **ABCONST.** This IOP model describes the water body by its total absorption and scattering coefficients and phase function; the water is taken to be homogeneous. This model is useful radiative transfer studies done at one wavelength. Output can be in optical or geometrical depth.

☐ **ABCASE1.** This IOP model uses simple bio-optical models to obtain the absorption and scattering coefficients from a user-specified chlorophyll concentration. This model is for use only in case 1 water.

☐ **ABAC9.** This model constructs the IOPs as the sum of pure water values plus "everything else"—particles and dissolved substances—as measured by a WETLabs ac-9 instrument. (Other data on the same format, e.g. from a WETLabs HiSTAR, can be processed with this model.)

☐ **OTHER.** This option allows you to select any user-written IOP model with up to ten components. See Section 6.1 for information on writing IOP models.

The next forms show the input required by these four IOP models.

The ABCONST form. This form requests the input needed by the ABCONST IOP model. The water is modeled as a single component having the specified total absorption, total scattering, and phase function. The pull-down menu shows the available phase functions. Even though the water IOPs are specified once a , b , and the phase function have been selected, the form also requests a wavelength for the run; the wavelength may be needed by the sky and/or bottom models. The code for the ABCONST model is in file `maincode\abconst.f`.

The ABCASE1 form. This form requests the input needed by the bio-optical models used in the ABCASE1 IOP model. The water is modeled by two components: pure water, and pigmented particles and co-varying CDOM. The absorption and scattering properties of the particles and CDOM are parameterized by the chlorophyll concentration, which in general is a function of depth. The needed chlorophyll concentration must be obtained either from a *user-supplied* subroutine or from a *user-supplied* file containing measured depth vs. Chl values. The `chlzfunc.txt` file in the template directory contains the format required for chlorophyll subroutines, and `chlzdata.txt` shows the format for chlorophyll data files. The code for the ABCASE1 model is in file `maincode\abcase1.f`.

The ABAC9 form. In this IOP model, the water is modeled by two components: pure water and “everything else” (particles and CDOM). Measured absorption and beam attenuation data as obtained from a WETLabs ac-9 (or similar) instrument is used to determine the IOPs of the second component. The pull-down menu allows you to pick the phase function to be used with the second component. The name of the file containing the ac-9 data must be entered. The code for the ABAC9 model is in file maincode\abac9.f.

The OTHER form. This form allows you to select a user-written IOP model with up to ten components. The subroutine on file maincode\abcas2.f is an example of such a model. First, enter the name of the file containing the subroutine (“abcas2” in this example), and the number of components used in the subroutine to build up the IOPs (4 for abcas2). Then select the phase function to be used for each component of your IOP model. In the present example, component 1 is pure water, so the pure water phase function on file pureH20.dpf is selected for component 1. Component 2 is “particles”, for which the Petzold “average particle” phase function is to be used. Component 3 is CDOM, which is assumed to be non-scattering; therefore any phase function can be used. Component 4 is mineral particles, for which a Fournier-Forand phase function is to be used. See Section 6.1 on writing your own IOP subroutines.

The INTERNAL SOURCE AND INELASTIC SCATTER form.

This form allows you to specify whether the run will include internal sources (e.g., bioluminescence) or various kinds of inelastic scattering processes. Except for Raman scattering, which is determined by water alone, choosing one of these features will require you to specify additional information. In the present example, checking the "CDOM fluorescence" box causes a pop-up notice to appear informing you that you must supply a subroutine named ACDOM (see Section 6.3). This subroutine will be called to get the CDOM absorption as a function of depth and wavelength, which must be known in order to compute the CDOM fluorescence. (This same routine is called by the abcase2 IOP model of the previous example to obtain the CDOM absorption.)

Note that if inelastic scatter is to be included in the run, the relevant excitation and emission wavelengths must be included (to be specified on the wavelength form). Therefore, the inelastic-scatter options are not available if you selected the ABCONST IOP model, which is for runs at only one wavelength.

The screenshot shows a dialog box titled "HYDROLIGHT" with the subtitle "Internal Source and Inelastic Scatter Selection". The main text asks the user to "Specify whether the run will include bioluminescence and/or inelastic scattering". There are two main sections: "bioluminescence" with a checked checkbox, and "Inelastic scatter" which contains three sub-options: "chlorophyll fluorescence" (unchecked), "CDOM fluorescence" (checked), and "Raman scattering" (unchecked). At the bottom are "Back" and "Continue" buttons.

The screenshot shows a dialog box titled "HYDROLIGHT" with the subtitle "Notice". The text informs the user that the "USER-SUPPLIED routine 'acdom' will be called to obtain the CDOM absorption as a function of depth and wavelength." It also states that "The run must include the relevant excitation and emission wavelengths." An "OK" button is at the bottom.

The WAVELENGTH SELECTION form. This form allows you to select the wavelength range and bandwidths for the run. In the present example, the option of making the run at a single wavelength is not available because CDOM fluorescence was included on the previous form. Suppose that we are interested only in the 400-600 nm range. The run is nevertheless started at 350 nm in order to include CDOM-fluoresced light that is excited in the near UV and emitted at blue wavelengths. As shown here, the wavelength bands for the run will be 350-370, 370-390, ..., 570-590, and 590-600 nm. If uneven bandwidths are desired (e.g., to exactly match the bandwidths of a particular sensor), the band boundaries can be entered manually, e.g., 400, 410, 425, 440, 450, Note that HYDROLIGHT must always run with contiguous wavelength bands. For example, you cannot make a single run with a band from 400-410 and another band from 435-445 nm without also including one or more bands in the 410-435 region.

HYDROLIGHT

Wavelength Selection

Specify whether the run will be at a single wavelength or over contiguous wavelength bands

☐ single wavelength

Enter the wavelength in nm (350 to 700):

☒ multi-wavelength

☒ Enter the wavelength minimum, maximum, and bandwidth

Enter the minimum wavelength in nm (≥ 350):

Enter the maximum wavelength in nm (≤ 700):

Enter the (uniform) bandwidth in nm (≥ 5):

☐ Enter a list of wavelength band boundaries (separated by commas):

The AIR-WATER SURFACE BOUNDARY CONDITIONS form. This form selects the wind speed to be used in modeling the roughness of the sea surface and the sky model to be used in modeling the incident sky radiance distribution. The “semi-empirical” sky model is recommended for routine use in oceanographic studies.

HYDROLIGHT

Air-Water Surface Boundary Conditions

Select the wind speed and a model for the incident sky radiance

Select a wind speed:

- ☐ 0 m/s (0 knots; 0 mph)
- ☐ 2 m/s (3.9 kt; 4.4 mph)
- ☒ 5 m/s (9.7 kt; 11.0 mph)
- ☐ 10 m/s (19.4 kt; 22.0 mph)
- ☐ 15 m/s (29.1 kt; 33.0 mph)

Select a sky model:

- ☒ use a semi-empirical sky model
- ☐ use an idealized sky model

The SEMI-EMPIRICAL SKY RADIANCE form. This form requests the input needed by the routines on files maincode\gcirrad.f and maincode\hcnrad.f. which together make the semi-empirical sky radiance model (see Section 6.4 for details). You can specify either the solar zenith angle or time and location information, from which the solar zenith angle will be computed by HYDROLIGHT.

The IDEALIZED SKY RADIANCE form. This form requests the input needed by the routines on files maincode\cosirrad.f and maincode\cosrad.f. which together make the idealized sky radiance model (see Section 6.4 for details). This sky radiance model is intended for radiative transfer studies that require a simple analytical model for the sky radiance distribution (such as a sun in a black sky or a heavy overcast). Note that there is no wavelength dependence of the sky radiance if this model is chosen.

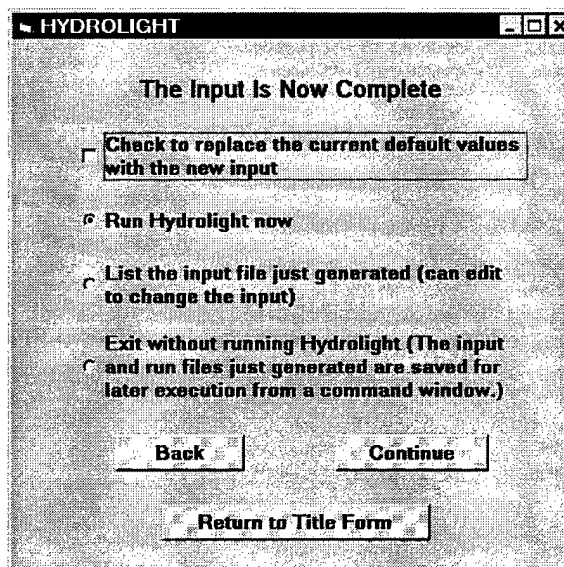
The BOTTOM BOUNDARY CONDITION form. This form requests the information needed to specify the bottom boundary condition. If the water is infinitely deep, the IOPs at the deepest depth of interest (to be specified on the output depth form) will be used to compute the reflectance of the infinitely deep layer of water below the region of interest. If the water is finitely deep, the irradiance reflectance of the bottom must be specified. In the option shown, the pull-down menu is used to select one of the available files of user-supplied reflectance vs. wavelength data (see Section 6.7).

The screenshot shows the 'Bottom Boundary Condition' dialog box in the HYDROLIGHT software. The title bar says 'HYDROLIGHT'. The main title is 'Bottom Boundary Condition' with the instruction 'Select the type of bottom boundary'. There are two radio buttons: 'is infinitely deep' (which is selected) and 'has a finite depth'. Below this, there's a section for 'The bottom reflectance is' with two options: 'Independent of wavelength' (selected) and 'dependent on wavelength'. The 'Independent of wavelength' option has a text input field for 'Enter the irradiance reflectance of the bottom in percent (0 to 100):' with the value '20' entered. The 'dependent on wavelength' option has a text area describing a user-supplied data file and a pull-down menu showing a list of files: 'Rgreenal.bot', 'Rcoralsd.bot', 'Rgreenal.bot' (highlighted), 'Rbrownal.bot', and 'Rredal.bot'. At the bottom are 'Back' and 'Continue' buttons.

The OUTPUT DEPTHS form. This form specifies the depths at which output from the run is to be saved for later analysis. Note that *the depths as which output is to be saved in no way affects the value or accuracy of the output at a given depth*. In the example shown, output is obtained at irregularly spaced depths near the surface and near the bottom, in order to get better depth resolution of the light field behavior near the surface and the bottom boundaries. However, if output were being obtained only at depths 0, 10, 20, and 25 m, the output at those four depths would be the same, but you would not have the additional values at the other depths. Note that the option of having the listed depths be optical depths is not available in this example, because we already have chosen to have a multi-wavelength run.

The screenshot shows the 'Output Depths' dialog box in the HYDROLIGHT software. The title bar says 'HYDROLIGHT'. The main title is 'Output Depths' with the instruction 'Select the depths where output is to be saved for post-run analysis'. A paragraph explains that output can be saved at up to 50 depths, and for a finite-depth water column, the last depth is the depth of the bottom. There are two radio buttons: 'enter a maximum depth and a depth interval' (selected) and 'enter a list of output depths (separated by commas; start with 0)'. The first option has input fields for 'maximum depth:' (25) and 'depth interval:' (5). The second option has a text input field containing the list '0, 1, 2, 5, 10, 15, 20, 22, 23, 24, 24.5, 24.9, 25.'. Below this, a section titled 'These depths are' has two radio buttons: 'geometric depths (in meters)' (selected) and 'optical depths (non-dimensional; single wavelength runs only)'. At the bottom are 'Back' and 'Continue' buttons.

The final form. All of the input required by HYDROLIGHT has now been obtained. If you check the REPLACE THE CURRENT DEFAULT VALUES box, the input you have just entered will be saved and will become the default values the next time you run the GUI. This will minimize the input required, if the next run is similar to the one now being made. If you click on continue, a command window (shown below) will be spawned and HYDROLIGHT will recompile (as necessary) and run in that window. After the command window is spawned, HYDROLIGHT is running



independently of the GUI. You can then click on RETURN TO RUN ID FORM and begin entering the input for the next HYDROLIGHT run. Note, however, that the next run cannot be started until the present run has finished. If you wish to check or change the input for the run, click on LIST THE INPUT FILE. The input file just generated (the file named Iroot.txt) will be shown. You can edit (e.g., to add another depth to the output) and re-save this file before making the run. If you are familiar with the format of the input file (described in Appendix A), this may be more convenient than passing through the entire GUI again in order to change the input.

Example command window showing that HYDROLIGHT has recompiled, relinked, and is running.

```

MRUGEx1 - MAINCODE
Auto
Compiling file KRADPATH.F.
Compiling program unit KRADPATH at line 1.
Encountered 0 errors, 0 warnings in file KRADPATH.F.

Lahey Fortran 90 Compiler Release 4.50e S/N: F9358786
Copyright (C) 1994-1998 Lahey Computer Systems. All rights reserved.
Copyright (C) 1985-1998 Intel Corp. All rights reserved.
Registered to: Curtis Mobley
                Sequoia Scientific, Inc.

Options:
-nap          -nbind      -nc             -nchk          -co
-ndal         -ndbl       -ndll          -exe maincode.exe
-nf90         -nfix       -g             -hed          -nin
-ninln        -nlist     -nls          -maxfatal 50  -nml
-o0           -no        -out maincode.exe
-sav          -stack 20000h -staticlink    -stchk        -nsun
-nsyn         -tp        -ntrace       -ntrap       -nvax
-un           -w         -nwin         -nwisk       -nwo
-wrap         -wref

386iLINK: 9.1fix1 -- Copyright (C) 1986-98 Phar Lap Software, Inc.
HYDROLIGHT IS RUNNING, PLEASE WAIT...
(To stop the run before normal termination,
 kill this command window. Output will be lost)

```

4.2 The Default Printout

Every standard HYDROLIGHT run generates an ASCII file of “printout” (the `Proot.txt` file in the `printout` directory) that is designed to be viewed with a text editor (or word processor). This file shows, in an easily read format, information such as

- The absorption, scattering, backscattering, and attenuation coefficients for each component of the IOP model
- The atmospheric parameters used in the sky radiance model
- Various irradiances (upward and downward scalar and plane irradiances)
- Radiances in selected directions (upward, downward, and horizontal at selected azimuthal directions relative to the sun’s position)
- Various apparent optical properties (mean cosines, reflectances, and diffuse attenuation functions)
- Quantities of interest in ocean-color remote sensing (incident and reflected sky radiance, water-leaving radiance)
- Any error messages generated during the run

Where appropriate, these quantities are tabulated at the user-defined depths where output was saved. The same information is repeated for each wavelength band.

It is strongly recommended that you always take a look at the printout after a HYDROLIGHT run terminates. This is easily done by opening the file with WordPad on Windows machines or with the text editor on UNIX machines; you can also use a word processor such as Word® or WordPerfect® to view the file. Scan the input to make sure it was what you intended. Inspect the output to see if it looks physically plausible. If you see something peculiar in the output, try to figure out the cause—incorrect input, lack of intuition on your part about underwater light fields, or (perish the thought) a bug in HYDROLIGHT. For some purposes, it may be convenient to copy output from the default printout and paste it into graphics or spreadsheet software for further analysis.

The default printout gives a reasonable amount of information for most users. You can obtain more or less printout by changing the appropriate parameter values in file `maincode\initial.f`, which contains the documentation needed to make such changes.

Note 1. The depths shown in the default printout may cause confusion. When you specify the depths were output is to be saved, HYDROLIGHT automatically adds a second depth just

below each depth you specify. These pairs of closely spaced depths are used to compute diffuse attenuation functions (K functions) by the appropriate finite difference approximation to the definition. For example, K_d , the diffuse attenuation coefficient for downwelling plane irradiance E_d , is estimated from

$$K_d \doteq -\frac{1}{E_d} \frac{dE_d}{dz} \approx -\frac{1}{0.5 [E_d(z_k) + E_d(z_{k+1})]} \frac{E_d(z_{k+1}) - E_d(z_k)}{z_{k+1} - z_k}.$$

The value of K_d so computed is the *average* value of K_d over the depth range z_k to z_{k+1} . However, the finite-difference approximation is an accurate estimate of K_d *at the midpoint* $\frac{1}{2} (z_k + z_{k+1})$ only if the two depths z_k and z_{k+1} are “close together,” even if the irradiances are computed with perfect accuracy. Therefore, for example, if you requested output to be saved at depths $z = 0, 1$, and 5 meters, HYDROLIGHT will actually compute and save the radiance and irradiances at depths $z = 0.00, 0.01, 1.00, 1.01, 5.00$, and 5.01 meters. The pairs $(0.00, 0.01)$, $(1.00, 1.01)$, etc. are used to estimate the depth derivatives. However, the printout is given only at the depths you requested, namely $0, 1$, and 5 meters in this example (these are the odd-indexed depths z_1, z_3, \dots . The even numbered depths z_2, z_4, \dots are used internally only for computing K functions and are not shown in the printout). These extra depths sometimes make the printout appear to be at every other depth (at the odd depths), unless you realize what is being done. The above example depths of 5.00 and 5.01 meters are used if the water column is infinitely deep with the last requested output depth being 5.00 m. If the water has a finite depth of 5.00 m, the last pair of depths is 4.99 and 5.00 m, in order that the bottom boundary be at exactly the requested 5.00 m depth.

Table 1 shows an example of the default printout for the irradiances and mean cosines. Note also that both the optical depth ζ (“zeta” in the printout) and the geometric depth z (in meters) are shown in the printout. Values in the air, just above the sea surface, are also shown where appropriate. Table 2 shows the corresponding printout for various K functions. The closely spaced pairs of depths (z_k, z_{k+1}) are shown in the printout as $(z_{\text{upper}}, z_{\text{lower}})$. The small increment used in computing derivatives, which is 0.01 m by default, is set in the routine `maincode\setdfits.f`. Picking depths only 0.01 m apart exceeds the ability of oceanographic instruments to measure the corresponding changes in the light field, but such closely spaced values gives excellent depth resolution of K profiles in the numerical model.

Irradiances (units of W/m ² nm), Mean Cosines (Mubars), and Irradiance Reflectance											
iz	zeta	z(m)	Eou	Eod	Eo	Eu	Ed	MUBARu	MUBARd	MUBAR	R = Eu/Ed
in air											
			9.7675E-02	9.5298E-01	1.0507E+00	2.9105E-02	5.9866E-01	0.2980	0.6282	0.5421	4.8616E-02
1	0.00	0.00	2.3508E-02	6.7162E-01	6.9513E-01	9.4290E-01	5.7898E-01	0.4011	0.8621	0.8193	1.6286E-02
3	1.91	1.00	1.1631E-02	3.0982E-01	3.2145E-01	4.4336E-01	2.5288E-01	0.3812	0.8162	0.7729	1.7532E-02
5	3.82	2.00	5.1328E-03	1.3393E-01	1.3907E-01	1.9295E-01	1.0722E-01	0.3759	0.8006	0.7571	1.7995E-02
7	9.56	5.00	3.8325E-04	9.8290E-03	1.0212E-02	1.4244E-04	7.7349E-03	0.3716	0.7869	0.7435	1.8415E-02
9	19.11	10.00	4.6379E-06	1.1866E-04	1.2330E-04	1.7210E-06	9.3190E-05	0.3711	0.7853	0.7418	1.8467E-02

Table 1. Example of the default printout showing various irradiances, mean cosines, and the irradiance reflectance. The line labeled "in air" gives the values just above the air-water surface. Depth 0.00 is in the water, just below the surface. The user-requested output depths were $z = 0, 1, 2, 5, 10$ m. These are the odd-indexed depths ($iz = 1, 3, 5, 7, 9$); the even depths are used to compute K functions and are not shown in the default printout.

K-functions (units of 1/meter) (valid only when zupper and zlower are closely spaced)									
zupper	zlower	z	Kou(z)	Kod(z)	Ko(z)	Ku(z)	Kd(z)	Knet(z)	KLu(z)
0.000	0.010	0.005	0.53271	0.69107	0.68571	0.65229	0.79845	0.80088	0.77136
1.000	1.010	1.005	0.78821	0.81967	0.81853	0.80976	0.84798	0.84866	0.83940
2.000	2.010	2.005	0.83970	0.85297	0.85248	0.84853	0.86592	0.86623	0.86168
5.000	5.010	5.005	0.87829	0.88019	0.88012	0.87955	0.88207	0.88212	0.88124
10.000	10.010	10.005	0.88430	0.88422	0.88423	0.88424	0.88410	0.88409	0.88406

Table 2. The default printout showing the K functions corresponding to the irradiances of Table 3. The closely spaced pairs of depths ($z_{\text{upper}}, z_{\text{lower}}$) used to compute the K functions are now shown explicitly.

4.3 Examining Output with Excel

A basic set of output is written to two ASCII files that are intended for conversion to Microsoft Excel® spreadsheets. This conversion is performed using Excel macros that are provided with HYDROLIGHT. These files contain roughly the same information as the default printout and, likewise, can be tailored to the individual user's desires. The two files, **Mroot.txt** and **Sroot.txt** (where "root" is the root name described in Sections 3.2 and 4.1), are written by HYDROLIGHT to the **excel** directory. They are structured as follows:

Mroot.txt will convert to an Excel workbook in which each sheet contains a single variable (such as the total absorption or the downwelling plane irradiance) as a function of depth and wavelength. This is the "multi-wavelength" format. This file is converted into a spreadsheet by running the macro **multiwl.xls**, which is in the **excel** directory.

Sroot.txt will convert to an Excel workbook in which each sheet contains several variables (such as the absorption, scattering, backscatter, and beam attenuation coefficients) as functions of depth, but grouped one wavelength at a time. This is the "single-wavelength" format. This file is converted into a spreadsheet by the **singlewl.xls** macro in the **excel** directory.

The way in which you run the macros depends slightly on the version of Excel you are using. The following example is for Excel 97 (Excel Version 7 follows the same procedure, but requires fewer steps because it does not give you the asinine warning messages). Suppose, for example, that you have made a multi-wavelength HYDROLIGHT run with the root name **UGEx1**. The **excel** directory should then contain files named **MUGEx1.txt** and **SUGEx1.txt**, as well as the macro files **multiwl.xls** and **singlewl.xls** (and any other **Mroot.txt** and **Sroot.txt** files from previous runs). To convert **MUGEx1.txt** to a workbook, follow these steps:

- start Excel
- select FILE → OPEN
- go to the **excel** directory (under the HYDROLIGHT main directory), select **multiwl.xls**, click OPEN

- click ENABLE MACROS if you get a warning message about the dangers of running macros
- click OK if you get a message about having to use Visual Basic to open macros
- select TOOLS → MACRO → RUN (select the macro named Macro_multiwave if it is not automatically highlighted)
- A pop-up box will ask for the name of the file to be converted. Enter MUGEx1.txt in the box (the full path might be c:\h40\excel\mugex1.txt, but only the file name is needed if your are in the excel directory)
- The macro will run and convert the Mroot.txt file into a workbook.
- Click NO if you get a message asking if you want the information on the clipboard to be available later
- The macro will ask if you want to save the workbook. If you answer YES, the default name for the workbook is MUGEx1.xls (Mroot.xls in general).

Whether or not you save the workbook, the macro leaves you in the Excel workbook that has been created. At this point you are simply using Excel without regard to how the workbook was created or to the fact that the data came from HYDROLIGHT. You can, for example, now block out data and use the Excel plotting functions to create simple plots of the HYDROLIGHT output. However, it is not the purpose of this users' guide to teach you Excel, so good luck and call the folks at Microsoft if you have questions.

Table 3 shows the first few rows of the first sheet of the workbook created in this example.

Run Title: Users' Guide, Example 1						
abs coef	18		7			
a (1/m)						
wavelen	0	5	10	15	20	25
360	6.03E-02	6.89E-02	8.73E-02	1.15E-01	9.70E-02	5.56E-02
380	5.08E-02	5.91E-02	7.69E-02	1.03E-01	8.63E-02	4.62E-02
400	4.44E-02	5.31E-02	7.16E-02	9.91E-02	8.14E-02	3.96E-02
420	4.99E-02	6.07E-02	8.37E-02	1.18E-01	9.59E-02	4.39E-02
440	5.35E-02	6.47E-02	8.87E-02	1.24E-01	1.01E-01	4.73E-02
460	5.16E-02	6.15E-02	8.26E-02	1.14E-01	9.37E-02	4.62E-02
480	4.82E-02	5.65E-02	7.42E-02	1.01E-01	8.36E-02	4.36E-02
500	4.99E-02	5.67E-02	7.12E-02	9.27E-02	7.88E-02	4.62E-02

Table 3. An example of an Excel work sheet created from an Mroot.txt file. There is one variable (here, the absorption coefficient a) displayed as a function of wavelength (rows) and depth (columns).

Creating a workbook from the single-wavelength file proceeds in the manner just described, except that you run the macro named singlewl.xls and you give it the name of a file on the single-wavelength format, e.g., SUGEx1.txt in the present example. Table 4 shows the first few rows of the first sheet of the workbook created from the example single-wavelength spreadsheet file.

Run Title: Users' Guide, Example 1

Wavelength = 360.0 nm

depth	a	b	c	omega0	bb
(m)	(1/m)	(1/m)	(1/m)		(1/m)
0	0.0603	0.3102	0.3705	0.8372	0.01146
5	0.0689	0.3794	0.4483	0.8463	0.01273
10	0.0873	0.5252	0.6125	0.8575	0.0154
15	0.1145	0.738	0.8525	0.8657	0.0193
20	0.097	0.6013	0.6983	0.8611	0.0168
25	0.0556	0.2717	0.3273	0.8302	0.01076

Wavelength = 380.0 nm

depth	a	b	c	omega0	bb
(m)	(1/m)	(1/m)	(1/m)		(1/m)
0	0.0508	0.2919	0.3427	0.8519	0.00988
5	0.0591	0.3575	0.4166	0.8581	0.01108
10	0.0769	0.4956	0.5725	0.8656	0.01361
15	0.1034	0.6972	0.8006	0.8709	0.0173
20	0.0863	0.5677	0.6541	0.868	0.01493
25	0.0462	0.2554	0.3016	0.8469	0.00921

Table 4. The first few rows of the first worksheet created from an example Sroot.txt file. Note that there are several variables (columns) shown as functions of depth (rows), with a separate block of data for each wavelength.

The FORTRAN routines that write the Mroot.txt and Sroot.txt files are found on file maincode\excel.f. Subroutine WRTXCLS writes the Sroot.txt files and subroutine WRTXCLM writes the Mroot.txt files. Those routines contain extensive documentation explaining how to modify the routines to obtain more or less output for conversion to Excel spreadsheets.

4.4 Plotting Output with IDL

Many HYDROLIGHT users prefer the IDL[®] software package for plotting output from HYDROLIGHT runs (IDL is Interactive Display Language, a product of Research Systems, Inc.). The `idl` directory distributed with the HYDROLIGHT code contains the routines used to plot Figures 4-8 of this Users' Guide; the IDL files are named `ugfig4.pro` to `ugfig8.pro`. If you are familiar with IDL (and have the IDL software on your computer), running these routines will be straightforward. If you do not want to use IDL, you do not need the files on the `idl` directory.

The IDL routine `readall.pro` (found on the `idl` directory) reads the HYDROLIGHT `Droot.txt` files. These large files contain all of the output from a HYDROLIGHT run and are, in fact, formatted for ease of reading into IDL. All of the information contained in a `Droot.txt` file is made available to the IDL main program via common blocks. A main IDL program can then generate the desired plots using the appropriate HYDROLIGHT output.

Note 1. There is a point of perpetual confusion when using IDL to plot HYDROLIGHT output. In HYDROLIGHT, an N -dimensional array $A(i)$ is usually (but not always) indexed from 1 to N , i.e. $A(1)$ to $A(N)$. In IDL, array indices always begin with 0, so that the same array is indexed as $A(0)$ to $A(N-1)$. This can be especially confusing when you are trying to extract blocks from multidimensional arrays in order to generate, for example, a polar-angle plot of the radiance in the azimuthal plane of the sun, at a given depth and wavelength. To make matters worse, HYDROLIGHT sometimes indexes arrays beginning with 0. For example, the arrays that store irradiances use depth index 0 to store the irradiance in the air, just above the surface. In this case, the HYDROLIGHT and IDL array indices are the same. Most IDL users have learned to live with these sorts of indexing mismatches, but if you have not, be forewarned and do some extra debugging of your IDL routines to make sure you are plotting exactly the HYDROLIGHT you want.

4.5 Setting Defaults

Many of the calculations performed within HYDROLIGHT assume default values for various parameters in order to minimize the input required from users for each run. These defaults can be simple numbers, such as the chlorophyll-fluorescence quantum efficiency; entire data sets, such as are used by the atmospheric models contained within the sky radiance models; or functions, such as the excitation-emission spectrum for CDOM fluorescence. In

most cases, the default values are set to typical or average values for the marine environment; these default values may or may not be adequate for your particular simulation. Also, defaults are used to set the amount and type of output obtained from a run.

There are a number of subroutines in the `maincode` directory where defaults are set. You should inspect these routines, and change the default values if they are not to your liking. Some of the places to look are as follows:

<u>filename</u>	<u>defaults set in the file</u>
<code>dimens.inc</code>	parameter values for array dimensions
<code>setdfits.f</code>	directory names for DOS or UNIX machines selects "Smith and Baker" or "Pope and Fry" pure water absorption values depth increment for computing depth derivatives for K functions basic atmospheric parameters for Gregg and Carder sky irradiance model overall amount of data saved for spreadsheet analysis miscellaneous other values
<code>initial.f</code>	sets the type and amount of printout on the <code>Proot.txt</code> file
<code>gcirrad.f</code>	additional parameters for the Gregg and Carder sky irradiance model
<code>excel.f</code>	sets the type and amount of data written to files <code>Mroot.txt</code> and <code>Sroot.txt</code> for Excel spreadsheet analysis

5. SPECIAL RUNS

A specialized HYDROLIGHT run is required in three circumstances: (1) a new phase function is to be prepared and added to the collection of available phase functions, (2) a new surface wind speed is needed, or (3) a new quad layout is needed. Because of the infrequent need for making specialized runs, control of these runs is not incorporated into the front-end programs. Specialized runs can be made only by creating the needed input file with a text editor and then submitting the run from a command window; this is easy to do.

5.1 Discretizing a Phase Function

The most common specialized run “discretizes” a scattering phase function $\tilde{\beta}(\psi)$ using the code in directory `discpf` (discretize phase functions). The discretization calculations average the phase function over all pairs of quads as described in *Light and Water*, Section 8.2; the main calculation is the evaluation of Eq. 8.13. The HYDROLIGHT code comes with several phase functions already discretized and ready for use. These phase functions are defined by FORTRAN subroutines in the `discpf` directory; the corresponding files of discretized phase functions are in the `..data\phasefun` directory and have names of the form `*.dpf`. Table 1 shows the discretized phase functions distributed with HYDROLIGHT.

Most users will sooner or later want to add to the collection of available phase functions. This is done as follows (note that all of these calculations are performed within the `discpf` directory):

Step 1. Write a FORTRAN subroutine to define the desired phase function. This can be done by modifying any of the distributed routines shown in Table 5 or by following the template on file `template\phasefun.txt`. Note that any scattering phase function $\tilde{\beta}(\psi)$ must satisfy the normalization condition

$$2\pi \int_0^{\pi} \tilde{\beta}(\psi) \sin\psi \, d\psi = 1 .$$

After the new subroutine has been written, give the file a name of the form `_name_.f` and save it as an ASCII (text) file in the `discpf` directory.

phase function (reference)	defining subroutine in directory discpf	file containing the discretized phase function in directory ..\data\phasefun
isotropic (<i>L&W</i> Eq. 5.105)	pfiso.f	isotrop.dpf
pure water (<i>L&W</i> Eq. 3.30)	pfpure.f	pureH2O.dpf
“Petzold” average particle (<i>L&W</i> Table 3.10, column 6)	pfpart.f	avgpart.dpf
One-Term Henyey-Greenstein (<i>L&W</i> Eq. 3.34)	pfothg.f (<i>g</i> is a user-defined parameter)	othgg90.dpf (for <i>g</i> = 0.90)
Fournier-Forand (<i>Ocean Optics XII</i> , SPIE vol 2258, p 194)	pfff.f (<i>n</i> and <i>nu</i> are user- defined parameters)	ff10540.dpf (for <i>n</i> = 1.05 and <i>nu</i> = 4.0) ff11540.dpf (for <i>n</i> = 1.15 and <i>nu</i> = 4.0)

Table 5. Phase functions distributed with HYDROLIGHT.

Step 2. The routine on file `phasef.f` will call your phase function subroutine during the discretization calculations. Therefore, you must change the subroutine call in file `phasef.f` to show the name of your phase function subroutine. For example, if the current call is

```
call pfpart(cospsi, pfvalue)
```

and you have named your subroutine “mynewpf”, you must change the above call to

```
call mynewpf(cospsi, pfvalue)
```

This change is made by opening file `phasef.f` with a text editor, making the change, and re-saving the file.

Step 3. The file `Input.txt` has two records that are read by the discretization program. The first record gives a descriptive title, which becomes the first record of the discretized phase function file. The second record is the root name to be given to the file containing the discretized phase function. An example of these two records is

```
Petzold "average particle" from L&W Table 3.10; standard quad partition
avgpart
```

The root file name ,`avgpart` in the above example, will be used to generate the name of the file containing the discretized phase function; this would be `avgpart.dpf` in the present example. The file extension “`dpf`” identifies the file as containing a discretized phase function. The `dpf` extension is only a convenient reminder of what is in the file; HYDROLIGHT does not actually require this naming convention for phase function files. *Edit the `Input.txt` file to change the title and file-name records as desired, and re-save the file.*

Step 4. Open a command window in the `discpf` directory. Enter the command

```
run
```

(or `run.bat`). This command runs the batch file `Run.bat`, which compiles the new phase function routine, creates an executable program named `discpf.exe`, and runs this program with the input on file `Input.txt`. The printout from the discretization run is written to file `Printout.txt` in the `discpf` directory. Inspect the printout (with a text editor) to verify that the discretization run terminated normally. If it did, then the new phase function is now available.

Note 1. If you are using the Lahey FORTRAN 90 compiler and a window titled “Automake Configuration File Editor” pops up when you run the discretization program, answer its questions as follows:

Compiler:	LF90 - Debug	(this is the default)
Compilation files:	*.f	(*.f90 is the default)
Target file:	discpf.exe	(Target.exe is the default)

The compiler and linker will now compile all of the FORTRAN code and create an executable file with the name expected by the `run.bat` file. See Section B.4 for details.

Note 2. The discretization program adds the file name, e.g., avgpart.dpf, to the end of the file named filelist.txt in the discpf directory. The filelist.txt file is read by the front-end program to generate the menu of available discretized phase functions. The files are shown in the order listed in filelist.txt. You can edit this file to change the order of the files in the menu, or to add or remove files from the available list.

Note 3. The main discretization program on file discpf\main.f sets several parameters used in the discretization. The default values are adequate for most phase functions and will not need to be changed by most users. However, this routine also sets directory names for either DOS or UNIX conventions. The default is for DOS; users running UNIX will need to change the default in main.f before running the program.

Note 4. The discretization calculations depend on the quad layout. If you wish to use a quad layout other than the default one shown in Fig. 2, you must re-create all of the files of discretized phase functions. This would be done after creating the new surface files as described in the next section.

5.2 Creating a New Surface File

The other type of specialized run creates a new file containing the discretized functions that describe the radiative transfer properties of the air-water surface for a given quad layout and wind speed. These functions are the quad-averaged versions of the four radiance transfer functions seen in *Light and Water* Eqs. (4.3) and (4.4). The calculations performed by the routines in directory surfcode (air-water surface code) are described in *Light and Water* Section 4.7; see in particular Eq. (4.74).

The distributed HYDROLIGHT code contains surface files for wind speeds of $U = 0, 2, 5, 10$ and 15 m s^{-1} ; these are the files named surfwind.0 to surfwind.15 in the data\surfaces directory. These wind-speed options are built in to the front-end programs. These files were created using the standard quad layout shown in Fig. 2. When making HYDROLIGHT standard runs, the quad layout information is read from the surfwind.U file.

There are two situations that require running the surfcode routines: (1) a new wind speed is desired, and (2) a new quad layout is desired. If the standard quad layout and selection of wind speeds is adequate for your purposes, you will never need to run the surfcode routines.

Although HYDROLIGHT uses numerically efficient invariant imbedding techniques to solve the radiative transfer equation within the water, the surfcode routines use Monte Carlo

ray tracing to estimate the four surface reflectance and transmittance functions. This brute-force numerical method is employed simply because it is the only mathematically tractable way to simulate the radiative properties of random sea surfaces.

In the event that a new surface file is needed, the surfcode routines are run as follows (note that all of these calculations are performed in the surfcode directory):

Step 1. Select a FORTRAN subroutine to define the desired quad layout. Several example quad-layout routines are distributed (in directory surfcode) with HYDROLIGHT; these are shown in Table 6.

quad layout	file name	generates the layout seen in
HYDROLIGHT standard 10x15 degree	std10x15.f	Fig. 2 of this report
equal solid angles	eqsang.f	<i>L&W</i> Fig. 4.19a
equal theta values	eqthet.f	<i>L&W</i> Fig. 4.19b and 4.19c
ad hoc theta values	adhoc.f	<i>L&W</i> Fig. 4.19d

Table 6. Distributed routines for generating quad layouts.

The documentation in these files explains how to define the desired θ layout. It is suggested that the user simply modify the `adhoc.f` file if a different layout is desired.

Step 2. The file `Input.txt` has two records that are read by the surfcode program. The first record gives a descriptive title, which becomes the first record of the discretized surface file. The second record is the wind speed U in m s^{-1} at an anemometer height of 12.5 m. This is the parameter used in the Cox-Munk capillary wave slope statistics; see *Light and Water* Eq. (4.32). An example of these two records is

```
Surface file for U = 5 m/s; standard quad partition
5.0
```

The wind speed will be used to generate the extension for the name of the file containing the surface information; this would lead to a file named `surfwind.5` in this example. *Edit the*

Input.txt file to change the title and wind-speed records as desired, and re-save the file.

Step 3. Edit the file *initial.f* to set the value of *iqpart* to select the desired quad layout subroutine (e.g., *iqpart* = 1 selects the standard quad layout; *iqpart* = 4 selects the ad hoc layout, etc.). See the documentation in this file for details.

Step 4. Open a command window in the *surfcode* directory. Enter the command

`run`

(or *run.bat*). This command runs the batch file *Run.bat*, which compiles the code, creates an executable program named *surfaces.exe*, and runs this program with the input on file *Input.txt*. The printout from the run is written to file *Printout.txt* in the *surfcode* directory. Inspect the printout (with a text editor) to verify that the run terminated normally. If it did, then the new surface file is now available.

Note 1. If you are using the Lahey FORTRAN 90 compiler and a window titled “Automake Configuration File Editor” pops up when you run the *surfcode* program, answer its questions as follows:

Compiler:	LF90 - Debug	(this is the default)
Compilation files:	*.f	(*f90 is the default)
Target file:	surfcode.exe	(Target.exe is the default)

The compiler and linker will now compile all of the FORTRAN code and create an executable file with the name expected by the *run.bat* file. See Section B.4 for details.

Note 2. Even though the new surface file has been created (say for a 7 m s^{-1} wind speed), this wind speed option will *not* automatically become available in the front-end programs. Adding a new wind speed to the GUI requires minor modification of the corresponding Visual Basic code in directory *guicode* and re-creation of the GUI executable. This is easily done if the user has Visual Basic® and some familiarity with programming in that language; if this is not the case, contact user support to obtain a modified GUI. An option for the new wind speed can be added to the text-based front end program by making a slight change in the corresponding FORTRAN program; see the documentation there.

Note 3. The only environmental, or physical, variable that must be specified explicitly when running surfcode is the wind speed U . However, the surface optical properties also depend on the real index of refraction, which is set to 1.34 in the `initial.f` subroutine. This routine also sets several other parameters set to defaults values, e.g. the number of rays to trace in the Monte Carlo simulation of a random sea surface, the parameters for the Cox-Munk slope statistics, and a switch for DOS (the default) or UNIX. Some users may need to change these values; see the documentation in routine `initial.f`.

WARNING 1. The HYDROLIGHT code as distributed has the standard quad layout with 20 θ (or $\mu = \cos\theta$) bands and 24 ϕ bands. If the user wishes to use a quad layout with more than 20 θ or 24 ϕ bands, then the array-dimension parameters seen in files `dimens.inc` in the `surfcode`, `discpf`, and `maincode` directories must be increased accordingly. If this is necessary, remember that there are M bands of quads in the θ direction, counting the two polar caps, and N bands in the ϕ direction. The value on `nmu` in `dimens.inc` is $M/2$, and the value of `nphi` is N . See the additional documentation in the `dimens.inc` file.

WARNING 2. If the quad resolution is changed from the default HYDROLIGHT standard layout seen in Fig. 2, then all surface files and discretized phase function files must be recreated. This is not a major undertaking, but it must be done because the distributed `data\surfaces\surfwind.U` and `data\phasefun*.dpf` files were discretized with the standard quad layout.

6. CREATING INPUT ROUTINES AND DATA FILES

HYDROLIGHT comes with several example subroutines for IOP models, sky models, analytic chlorophyll profiles, etc.; and it comes with example data files of measured chlorophyll profiles, bottom reflectances, and the like. You can use the example models as they are, and in some cases this is appropriate. However, most users sooner or later will want to replace the example IOP models with models tailored to specific water bodies. In all cases, you must replace the example files of chlorophyll data and ac-9 data with your own.

This section describes how to write your own subroutines for various models, and how to format various data files so that they can be read directly by the HYDROLIGHT routines. Writing a subroutine for an IOP model, for example, does require some knowledge of FORTRAN. However, HYDROLIGHT comes with templates that show you how to “fill in the blanks” to create IOP and other models, and how to format data files, so this process is relatively easy even for inexperienced programmers. The best way to learn how to write a subroutine or format a data file is to compare the appropriate template file (all of which are on the `template` directory) with the corresponding example subroutine or data file. The following sections give a few additional words of advice on writing your own routines.

6.1 Subroutines for IOP Models

The most common and important task in tailoring HYDROLIGHT to a specific water body is writing a subroutine to provide HYDROLIGHT with the IOPs—namely the absorption (a) and scattering (b) coefficients—of the water body. The HYDROLIGHT software package comes with five example “ab” subroutines for computing a and b . These routines, which are in the `maincode` directory, are described in Table 7. The `abconst`, `abcase1`, and `abac9` routines are suitable for use in a wide variety of studies. The `abpure` routine is just an example of a very simple IOP model: pure water only. The `abcase2` routine is a somewhat contrived example of how to write a multi-component (pure water, particles, and CDOM) IOP model; *this model is not intended for general use in case 2 waters.*

file name	components	comments
abpure.f	1: pure water	returns the a and b values for pure water
abconst.f	1: the total a and b values	returns the total a and total b values entered when running the front-end program; a and b are independent of depth
abcase1.f	1: pure water 2: particles and co-varying CDOM	for use in case 1 waters only; requires the chlorophyll concentration as input
abcase2.f	1: pure water 2: particles and co-varying CDOM 3: non-co-varying CDOM	a contrived example to show how to write a 3-component IOP model
abac9.f	1: pure water 2: everything else, as measured by an unfiltered ac-9	designed to read HYDROLIGHT standard format ac-9 (or similar) data files

Table 7. The example “ab” subroutines provided with HYDROLIGHT.

When writing your own IOP model, the main thing to remember is that the total IOPs of a water body are built up as a sum of IOPs attributable to the various components of the water body. Thus, the total absorption coefficient is computed from (*Light and Water*, Eq. 3.10)

$$a_{\text{total}}(z, \lambda) = \sum_{i=1}^{ncomp} a_i(z, \lambda).$$

Here $a_i(z, \lambda)$ is the absorption coefficient of the i^{th} component of the water body, which in general is a function of the depth z and wavelength λ . A similar equation is used to compute the total scattering coefficient b . The number of components in the IOP model is $ncomp$. The default array dimensioning in HYDROLIGHT allows $ncomp$ to as large as 10 (if you need more than 10 components, increase the `mxcomp` parameter in the `dimens.inc` file).

The process of writing your own *ab* subroutine is straightforward:

- **Decide what components will be included in the model, and in what order.** For example, you might let component 1 be pure water, component 2 be CDOM, component 3 be pigmented particles, and component 4 be mineral particles. The order of the components is not important in building up the absorption and scattering coefficients, but it is very important when you later specify (when running the front-end program) which scattering phase function is to be used for each component. If component 1 is chosen to be pure water, then you must select the phase function for pure water for component 1; if component 4 is mineral particles, then you must select a phase function for component 4 that is appropriate for mineral particles, and so on.
- **Decide how you are going to compute the absorption and scattering coefficients for each of your components.** For example, you can just call the HYDROLIGHT routine `pureH2O` to obtain the pure-water a and b values. You might want to use a simple formula to model the CDOM absorption coefficient and assume that CDOM is non-scattering. You might want to call HYDROLIGHT subroutine `chlzdata` to read in a file of your chlorophyll data, and then use a bio-optical model to convert the chlorophyll value at depth z into the component a and b values at z and λ . You might want to read your own data file of mineral particle a and b values. Such decisions are yours to make.
- **Modify one of the existing IOP subroutines, or fill in the `ab.txt` template, to code your IOP model as a subroutine suitable for calling by HYDROLIGHT.** The documentation in `template\ab.txt` explains how this must be done.

As explained in the `ab.txt` template, you must give your IOP subroutine a unique name, e.g., “myIOPs” or “abmodel5” and save it as a file with the corresponding name, e.g., `myIOPs.f`, in the `maincode` directory. Your model for a and b is then ready for use by HYDROLIGHT.

There are several important points to note about the “*ab*” subroutine just described:

- *An `ab` subroutine gives HYDROLIGHT information about the absorption and scattering coefficients, but not about the corresponding phase functions.* The phase function information must be entered when running the front-end program; this is why

it is important to remember the order of the components in the IOP model. The phase function to be used with each component is not built into the subroutine for a and b because users often want to change the phase function being used to model a particular component. For example, you may wish to have a suite of phase functions, each with a different amount of backscatter, to use in modeling a particle component. Such flexibility is a powerful features of HYDROLIGHT.

- *You can do whatever you wish within the ab subroutine to compute the component a and b values.* HYDROLIGHT requires only that the format of the call to the subroutine be fixed (i.e., the list of variables must be exactly as shown in the `ab.txt` template file).
- *The ab subroutine must return values for any possible depth and wavelength that might be used in a HYDROLIGHT run.* If you have data at discrete depths or wavelengths, you must use an interpolation or extrapolation scheme to define the a and b values at all other depths and wavelengths. HYDROLIGHT calls the ab routine thousands of times and at very closely spaced depths (not just at the depths were output is to be saved) when solving the radiative transfer equation.
- *HYDROLIGHT uses the absorption and scattering coefficients at the wavelength band centers, not band averages.* If band-averaged a and b values were used, the band averaging would have to be done each time the ab subroutine is called at a new depth; this would be computationally prohibitive. Using a and b at the band centers is not generally a large source of error because a and b vary slowly and smoothly on a scale of 1 to 20 nm. If your a and b do vary rapidly with wavelength, then use smaller band widths.

To run HYDROLIGHT with your IOP model, you select the OTHER IOP-model option when running the front-end program. Then enter the name of your routine and the number of components. The front-end program will then prompt you for the name of the file containing the discretized phase function to be used for each component. If you have named your ab model “myIOPs”, for example, the front-end program will create an “include” file named `abscat.inc`, which has the following call to your subroutine:

```
call myiops(z,wavenm,ncomp, acomp,bcomp,atotal,btotal)
```


The `abscat.inc` file is “included” in the HYDROLIGHT core routines wherever HYDROLIGHT needs to know the absorption and scattering coefficients for a given depth and wavelength.

6.2 Subroutines for Chlorophyll Distributions

It is often convenient in the “ab” models just discussed to be able to call a subroutine that returns the chlorophyll concentration *Chl* in mg Chl m^{-3} given the depth in meters. The chlorophyll concentration then can be used as input to bio-optical models for *a* and *b*, as is done in the subroutine on file `abcase1.f`, for example. A chlorophyll subroutine will always be called to obtain *Chl* if chlorophyll fluorescence is included in the HYDROLIGHT run.

Two such subroutines are provided with HYDROLIGHT. The first, found on file `maincode\chlzfunc.f`, uses an analytic function to compute the *Chl* concentration at any depth. This is the subroutine called if you select the USER-SUPPLIED SUBROUTINE option for the ABCASE1 IOP model when running a front-end program. *The simple formula on file chlzfunc.f is only an example; you must replace this formula with one that describes your own situation.* The second chlorophyll subroutine is on file `maincode\chlzdata.f`. This is the subroutine called if you select the USER-SUPPLIED DATA FILE option for the ABCASE1 IOP model. This routine reads a file of measured depth vs. *Chl* data, fits a spline to the data, and uses the spline to compute *Chl* at any depth. The HYDROLIGHT standard format for the data files read by `chlzdata.f` is seen on `template\chlzdata.txt`.

These two routines should be adequate for most purposes, after you replace the formula or the data file to describe your particular situation. If you do wish to write your own chlorophyll subroutine, follow the template on file `template\chlzfunc.txt`, or modify either of the two distributed routines.

6.3 Subroutines for CDOM Absorption

Just as for chlorophyll, it is often useful to have a subroutine that returns the CDOM absorption coefficient as a function of depth and wavelength. Such a routine will always be called by HYDROLIGHT if CDOM fluorescence is included in the run. The example case 2 water IOP model on file `abcase2.f` also calls a CDOM absorption routine.

One example CDOM absorption routine is included with HYDROLIGHT; it is on file `maincode\acdom.f`. This routine uses a simple formula to compute the absorption coefficient of CDOM given the depth and wavelength. *The particular formulas seen on file acdom.f are*

just for example purposes; you must replace these formulas with ones that describe your own situation. Rather than use analytic formulas, you can also modify `acdom.f` to read a data file of CDOM absorption values. Such a data file might be generated by sampling with filtered and unfiltered ac-9 instruments. In either case, note that the `acdom` routine must return the absorption value for *all* depths and wavelengths relevant to the HYDROLIGHT run.

6.4 Subroutines for Sky Models

HYDROLIGHT must know the sky radiance incident onto the sea surface for all directions and all wavelengths (that are relevant to the HYDROLIGHT runs to be made). In order to increase flexibility in using modeled or measured sky radiances, this information is provided by two independent subroutines. The first subroutine returns the direct and diffuse components of the sky *irradiance*. These irradiances are used to set the magnitude of the sky radiance. The second subroutine returns the angular pattern of the sky radiance distribution. This radiance pattern is integrated over direction to compute a sky irradiance, which is then forced to equal the irradiance given by the irradiance subroutine. Together, these two routines yield a sky radiance distribution with the desired magnitude in all directions.

HYDROLIGHT comes with two versions of each routine, as shown in Table 8.

file name	type	front-end option	comments
gcirrad.f	irradiances	SEMI-EMPIRICAL SKY MODEL	computes irradiances using Gregg and Carder (1990)
hcnrad.f	radiance	SEMI-EMPIRICAL SKY MODEL	computes the radiance pattern using Harrison and Coombes (1988)
cosirrad.f	irradiances	IDEALIZED SKY MODEL	uses user input from the front-end program to compute the irradiances
cosrad.f	radiances	IDEALIZED SKY MODEL	computes radiance patterns for uniform or heavy overcast skies by L & W , Eq. (4.50)

Table 8. The sky irradiance and radiance models distributed with HYDROLIGHT. The front-end option column shows which routines are called when either of the sky model options is selected when running a front-end program.

The two routines that are selected when the SEMI-ANALYTICAL SKY MODEL option is selected in the front-end program do a sufficiently good job of modeling the sky radiance for most applications of HYDROLIGHT. The use of this sky option is therefore recommended for use in general oceanographic studies. The two routines that are selected when the IDEALIZED SKY MODEL option is selected in the front-end program are intended for specialized radiative transfer studies that need a simple sky radiance distribution, e.g. a point sun in a uniform background sky, or a heavily overcast sky.

The front-end programs do not have an option for entering user-defined sky irradiance or radiance models, because the models provided are adequate for most purposes. However, you can write your own sky irradiance or radiance-pattern models by following the corresponding templates found on files `skyirrad.txt` and `skyrad.txt`. If this is done, you will need to change the corresponding include files in the main directory, `skyirrad.inc` and `skyrad.inc`, to show the names of your subroutines. These files will be over-written each time the front-end program is run, unless minor modification is made to the front-end program. Consult user service for advice if this is necessary.

6.5. Subroutines for Bioluminescence

If you check the BIOLUMINESCENCE option when running the front-end program, HYDROLIGHT will call the subroutine found on file `maincode\s0biolum.f` to compute the bioluminescence source function S_0 as a function of depth and wavelength. The function $S_0(z, \lambda)$ computed by the `s0biolum` routine is discussed in *Light and Water*, Section 5.16. As always, *the particular formulas see in file s0biolum.f are just examples of how to compute S_0 ; each user must replace these formulas with ones that describe the internal source distribution of interest.*

6.6 Data Files for WETLabs ac-9 Data

A common use of HYDROLIGHT is to compute light fields using absorption and scattering coefficients as measured by a WETLabs ac-9 (or HiSTAR) instrument as input. The ABAC9 IOP option is designed for precisely this task. When using the code on file `abac9.f`, the IOPs are modeled as a two-component system: component 1 is pure water and component 2 is "everything else," namely the particles and dissolved substances detected by the ac-9 (after pure water values are subtracted from the measured totals according to the factory calibration scheme). For many waters, the measured signal can be attributed primarily to

particles, and the scattering can be modeled with a particle-type phase function. This is assumed to be the situation when selecting the ABAC9 option. Equating “everything else” to “particles” will be inadequate in waters with high concentrations of CDOM, in which case the particle and CDOM signals can be separated by sampling with filtered and unfiltered ac-9 instruments. If such data are available, a corresponding three-component (pure water, particles, CDOM) IOP model should be written to process the particle and CDOM data sets separately.

In any case, please note that it is not HYDROLIGHT’s job to process your raw ac-9 data (or any other data). Before giving the ac-9 data to HYDROLIGHT, you must perform all of the usual tasks of checking for bad data, applying scattering corrections, and smoothing or binning the data with depth to remove as much instrument noise as possible. Recall the comments associated with Fig. 1. Once the data have been cleaned up, they can be placed on a file in the HYDROLIGHT standard ac-9 data format, which is seen on the file `template\ac9data.txt`. The file can then be given to HYDROLIGHT via entering the file name when running the front-end program; HYDROLIGHT by default looks in the `data` directory for files of ac-9 data.

When HYDROLIGHT reads a file of ac-9 data, it performs a number of initialization calculations:

- The a and c values are read for the discrete depths and wavelengths
- Minor error checking is performed. For example, if the depths as read are not in increasing order, the depth records are re-ordered; values with duplicate depths are averaged. However, such quality control should be done before giving the data to HYDROLIGHT
- $b = c - a$ is computed for each depth and wavelength
- the discrete-depth, discrete-wavelength a and b values are fit with bi-cubic splines

On all subsequent calls, the bi-cubic splines are used to define a and b at any depth and wavelength.

Note. Routine `abac9` reads in a and c , and then compute a and b . If you would rather convert a and c to a and b during your own data processing, just rewrite the `abac9` routine slightly to read in a and b , and omit the conversion calculations.

WARNING. If HYDROLIGHT needs values for a and b at depths or wavelengths outside the range of the original ac-9 data, the nearest measured values will be used (rather than terminating the run with an error message or using the splines to extrapolate, which can cause large errors). Thus you *can* start a HYDROLIGHT run at 350 nm even if your ac-9 data starts at 412 nm, but you *cannot* reasonably expect to get accurate results at much less than 412 nm. If this way of extrapolating the data beyond its measured range is not satisfactory, rewrite the abac9 routine to do it your way.

6.7 Data Files for Bottom Reflectances

HYDROLIGHT can simulate both infinitely deep and finite-depth water bodies. Let z_{\max} be the "maximum depth of interest" in a run, i.e., the maximum depth at which we wish to obtain output from the model. For a study in the open ocean, we might take $z_{\max} = 50$ m, even though the water is optically infinitely deep. In this case, the model assumes that the water is homogeneous below depth z_{\max} and has the same IOPs as are given by the ab routine at $z = z_{\max}$. (Note that the water column between depth 0, the mean sea surface, and depth z_{\max} generally has depth-dependent IOP's.) The model automatically computes the bi-directional radiance reflectance of the infinitely deep, homogeneous layer of water below depth z_{\max} and applies this reflectance as the bottom boundary condition at depth z_{\max} . Section 9.5 of *Light and Water* describes these calculations. Note that the infinitely deep layer of water below depth z_{\max} is not a Lambertian reflector.

In the finite-depth case, a physical bottom is placed at depth z_{\max} . The physical bottom is taken to be an opaque, Lambertian reflecting surface whose *irradiance* reflectance is specified by the user. Equation (4.81) of *Light and Water* then gives the needed bi-directional radiance reflectance of the physical bottom.

The irradiance reflectance for a physical bottom is most easily communicated to HYDROLIGHT by placing measured wavelength-dependent reflectances on a file in the HYDROLIGHT standard format for bottom reflectances and then giving the file name to the front-end program; by default, HYDROLIGHT looks for files of bottom data in the data\botmrefl directory. The steps to creating a file of bottom reflectance data are as follows:

- Place the wavelength vs. reflectance data on a file with the format seen in the file template\rbottom.txt.
- Give the file a meaningful name, e.g. Rmud.bot, and place the file in the data\botmrefl directory.

- Use a text editor to add the name of your file to the `filelist.txt` file in the `data\botmrefl` directory. The front-end programs read `filelist.txt` to get the names of the available files of bottom reflectances. The files are shown in the order listed in `filelist.txt`. You can edit this file to change the order of the files in the menu, or to add or remove files from the available list.

If you wish to use an analytical formula to compute the bottom reflectance given the wavelength, you can rewrite routine `maincode\bottom.f` to use your formula, rather than read a data file.

Note. HYDROLIGHT must know the bottom reflectance for any wavelength relevant to a run. If the bottom reflectance is obtained from a standard-format file of discrete-wavelength reflectance data, routine `maincode\bottom.f` uses a spline function to interpolate between the discrete wavelengths where the reflectance was measured.

7. ADVICE ON RUNNING HYDROLIGHT

This section gives a few words of advice that may help you run HYDROLIGHT more efficiently.

7.1 Depth Considerations

The relevant measure of depth in radiative transfer theory is the dimensionless *optical depth* ζ , not the geometric depth z . This can lead to confusion in understanding HYDROLIGHT run times. For example, in homogeneous clear water at blue wavelengths, where the beam attenuation is $c \approx 0.02 \text{ m}^{-1}$, 50 m is only about $\zeta = cz = (0.02 \text{ m}^{-1})(50 \text{ m}) \approx 1$ optical depth. In highly absorbing or scattering water with $c = 1 \text{ m}^{-1}$, 50 m corresponds to 50 optical depths. In this case, HYDROLIGHT will require ~50 times as long to obtain a solution to the same geometric depth as it does in the clear water. This dependence of run time on optical depth causes an unavoidable run-time penalty when running HYDROLIGHT at red or near-IR wavelengths (out to 800 nm), where the absorption by pure water itself is as large as 2.5 m^{-1} . Runs to even a few meters of geometric depth require the solution of the RTE to many optical depths at long wavelengths, even in pure water. Thus, much of the total run time can be expended in solving the RTE at the longer wavelengths. Therefore, do not run HYDROLIGHT to longer wavelengths than are really necessary to solve your problem. If you do need output for wavelengths out to 800 nm, consider making two separate runs: the first covering wavelengths below 700 nm and going down to the needed geometric depth, and the second from 700 to 800 nm but stopping at a shallower geometric depth (since there will be little light below the first few meters of the water column). Then piece the two solutions together.

There is a very useful trick that can enormously speed up the HYDROLIGHT run time in one special situation. *If (and only if) you are modeling a homogeneous, infinitely deep water body and the only output you need is the water-leaving radiance, then you can apply the bottom boundary condition at a nominal depth (say, 0.1 m).* (That is, request output only at depths 0.0 and 0.1 m and select the “infinitely deep water” bottom boundary condition.) The water-leaving radiances will be *exactly* the same as if you had solved the RTE to a greater depth, but the run time will be minimal. If you use this trick, you will of course not have any output for the light field below 0.1 m within the water, but that is irrelevant for remote-sensing studies. Note that the water-leaving radiances will not be the same in these “deep” and “shallow” simulations if the water IOPs vary with depth.

HYDROLIGHT automatically solves the RTE with extremely fine depth resolution in the IOP's. The numerical algorithms generally take very small depth steps ($\sim 10^{-6}$ optical depths or less) when solving the Riccati equations (*Light and Water*, Eqns. 8.74-8.85) that lie at the core of the HYDROLIGHT calculations. These calculations typically require about 90 per cent of the run time. As already noted, the primary factor influencing the run time is the optical depth to which a calculation is performed. The run time is much less influenced by the depth dependence of the IOPs. Thus, for example, solving the RTE for the binned absorption profile seen in Fig. 1 (the left-hand curve) does not require much more time than solving the RTE for a homogeneous water body with the same average absorption profile. (This statement does have its limits, though, as discussed in Section 1.1.) This near independence of run time on IOP depth structure is one of the most powerful features of HYDROLIGHT.

HYDROLIGHT *saves* the computed radiances only at the *user-selected output depths* z_k , $k = 1, 2, \dots, K$, which are specified along with the other input to a standard run. The output depths can be arbitrarily spaced in order to get detailed output in the regions of greatest interest, such as near the sea surface or near strong gradients in the IOP's, and less output in regions where there is little "fine structure." *It is very important to note that the solution of the RTE is entirely independent of where output is to be saved for later analysis.* For example, you might request output at one-meter depth intervals, i.e., $z_1 = 0$, $z_2 = 1.0$ m, $z_3 = 2.0$ m, ..., $z_6 = 5.0$ m, ..., and so on. On the other hand, you might request output at $z_1 = 0$, $z_2 = 5.0$ m, ..., and so on. The output at 5.0 will be exactly the same in both cases; the only difference is that we will have more output between 0 and 5.0 m in the first case. In particular, HYDROLIGHT does *not* solve the RTE with "one-meter depth resolution" in the first case and with "five-meter resolution" in the second case. The number of depths where output is saved does not significantly affect the run time (the only difference is the small amount of time required to write the output files). The size of the output files is, of course, directly proportional to the number of depths where output is saved.

The default array dimensioning supports output at $K = 50$ user-specified depths (strictly speaking, at 50 pairs of closely spaced depths, as described in Section 4.2). The parameter *mxz* in the *dimens.inc* files can be changed throughout the code if output at more depths is needed.

Most oceanographers want output at *geometric* depths z_k , measured in meters, for ease of comparison with observational data. However, in some cases the output depths also can be specified as dimensionless *optical* depths ζ_k . This option is available only if the model is being run at just one wavelength, because the wavelength dependence of the IOP's makes a given optical depth correspond to different geometric depths (different physical locations in

the water column) at different wavelengths. Computations in terms of optical depths are convenient for general monochromatic radiative transfer studies, as opposed to specific oceanographic studies. Note that if the model is being run with optical depths, then the “ab” routine must be written to accept optical depth as input; of the example routines distributed with HYDROLIGHT, only `abconst` accepts both optical and geometric depths.

7.2 Wavelength Considerations

The various data sets built in to HYDROLIGHT give it the ability to run anywhere in the wavelength domain 350-800 nm, which is of interest in optical oceanography.

There are two basic options for specifying the needed wavelength information. The first is to run HYDROLIGHT at a single wavelength. HYDROLIGHT then solves the *monochromatic* RTE with the IOP's, sky radiances, and bottom reflectance being taken equal to their values at the specified wavelength. Some routines, for example the “semi-analytical” sky model, have built-in data that are accurate to 1 nm resolution. Other routines, for example the absorption and scattering coefficients for pure water, may have data built in at 5 nm or 10 nm resolution. Those routines linearly interpolate if necessary to obtain IOP's at the requested wavelength. A run at a single wavelength can include an internal source term at that wavelength. Such a source term can represent a bioluminescing layer, for example. The output depths can be either geometric or optical depths. Monochromatic runs are most useful for general radiative transfer studies.

The second option is to run HYDROLIGHT over one or more contiguous wavelength bands with bandwidths $\Delta\lambda_j$, $j = 1, 2, \dots, J$. When this option is chosen, the model automatically averages the input sky radiance over each wavelength band. This averaging smooths out the large nanometer-to-nanometer fluctuations in the sky radiance magnitude owing to Fraunhofer lines in the solar spectrum. However, *the absorption and scattering coefficients are taken to be the values at the band centers*. In principle, $a(z, \lambda)$ and $b(z, \lambda)$ also should be averaged over each band. However, this averaging would have to be performed every time the “ab” IOP subroutine is called with a new depth during the solution of the RTE; such averaging would be an enormous computational expense. If the band widths are of size $\Delta\lambda \leq 20$ nm, then the replacement of band-averaged IOP values by band-center values will be acceptably accurate for most purposes, since IOP's do not fluctuate wildly on a nanometer scale.

Most present-day oceanographic and remote-sensing sensors have bandwidths of 5-20 nm. The bandwidths of a HYDROLIGHT run can be matched to these sensor bandwidths as

desired. There is no requirement that the bandwidths $\Delta\lambda_j$ be equal for different j values. If the user wishes to simulate a wider-bandwidth instrument (such as the Landsat Thematic Mapper), then HYDROLIGHT should be run with a number of smaller bandwidths. The final output for the smaller bandwidths then can be averaged to get band-averaged output for the wide band. This approach properly accounts for the wavelength variation of the IOP's within the large band.

Another consideration in the choice of wavelength bands arises in connection with inelastic scattering effects. Suppose, for example, that we wish to simulate the light field in the 450-500 nm region. If we are uninterested in inelastic-scattering effects from shorter wavelengths, then we could run HYDROLIGHT with five bands chosen as 450-460 nm, 460-470 nm, ..., 490-500 nm, for example. However, if we wish to include the contributions of fluorescence or Raman scattering to the light field in the 450-500 nm region, then HYDROLIGHT must be run for all wavelengths less than 450 nm, for which there might be an inelastic-scattering contribution to the region of interest. Thus to include Raman scatter, the model should be run starting with a band from 390-400 nm, since wavelengths near 400 nm will Raman scatter into wavelengths near 450 nm. If CDOM fluorescence is to be included, then HYDROLIGHT should be run starting at 350 nm, because CDOM fluorescence can be excited by ultraviolet wavelengths and because CDOM fluoresces throughout the visible.

7.3 Inelastic Scattering and Bioluminescence

HYDROLIGHT 4.0 has the option of running with or without inelastic scattering and internal sources being included in the RTE. The inelastic scattering processes included in model are chlorophyll fluorescence, CDOM fluorescence, and Raman scattering. The internal source usually is tailored to represent bioluminescence. If these effects are all omitted from the run, then HYDROLIGHT carries out a sequence of independent solutions of the monochromatic, source-free RTE. The solutions for different wavelength bands are then completely independent. However, if one or more of these effects are included, then the appropriate source terms are automatically added to the RTE, as described in *Light and Water*, Sections 5.14-5.16 and 8.7. In the case of inelastic scattering, the solutions in different wavelength bands are coupled by the inelastic scattering from shorter to longer wavelengths. The inelastic-scattering and internal-source computations in some cases call upon user supplied routines, as noted in Sections 6.2, 6.3, and 6.5.

Function subroutine `acdom(z,λ)` returns the absorption by CDOM (in units of m^{-1}) at any depth and wavelength. The computation of CDOM fluorescence requires this information.

As with `chlz`, routine `acdom` also can be used for other purposes such as the computation of total absorption; recall the version of `abscat` given on file `abcase2.f`. File `acdom.f` gives an example of `acdom`. HYDROLIGHT models CDOM fluorescence using the spectral fluorescence quantum efficiency function of *Light and Water* Eq. (5.101), as shown in *Light and Water* Fig. 5.11. This particular function is built in to routine `wrfcdom` on file `wrfdisc.f`. The user can replace this default function with another, if desired.

Because Raman scattering depends only on the water itself, it is always the same and no user-supplied information is required. Various parameter values, such as the Raman cross section, are set to default values in routines `shatram` on `shatram.f` and `wrframen` on file `wrfdisc.f`. Those values can be changed if desired.

Some additional computational expense results from the addition of the source terms seen in Eqs. (8.74)-(8.85) of *Light and Water*. However, the main increase in computation time when inelastic scattering is included arises from the need to run the model over wavelengths shorter than the wavelengths of interest, as was illustrated in the previous subsection. In the example discussed there, where the interest was only on 450-500 nm, including CDOM fluorescence effects on the 450-500 nm band requires about three times the computational expense, because the model then must be run from 350-500 nm.

The HYDROLIGHT 4.0 model includes chlorophyll and CDOM fluorescence and bioluminescence exactly as formulated in Sections 5.15 and 5.16 of *Light and Water*. However, Raman scatter is included using an azimuthally averaged effective source term that is equivalent to the formulation seen in Appendix A of Mobley, et al. (1993). This simplification allows the Raman effective source term to be computed from the scalar irradiance (as is the case for fluorescence), rather than from the full radiance distribution. The azimuthally averaged formalism yields the correct Raman contribution to irradiances, which are computed from the azimuthally averaged radiance. However, the Raman contribution to the radiance is correct only as an azimuthally averaged value.

8. EXAMPLE HYDROLIGHT RUNS

This section briefly describes several example HYDROLIGHT runs. The input and output files for these examples are all found on the **examples** directory. (These files were moved from their respective directories as described in Section 3.3 to **examples** after making the runs.) You can reproduce these runs on your own computer and compare your output with that from the example runs to verify that HYDROLIGHT is performing correctly on your computer. The example runs were all made on a 400 MHz Pentium Pro processor. Run times (shown in the printout) on your computer will be longer or shorter, depending on the speed of your computer. (Note also that the “run time” given in the printout is “wall clock” time, not true computation time. Therefore, identical runs may show different run times in the printout, if the computer also was performing other tasks while HYDROLIGHT was running.) The output from your runs should be identical to that from the examples to within numerical roundoff errors, which may cause very slight differences in some output values.

8.1 Discretizing a Phase Function

The step-by-step process is described in Section 5.1. The **discpf** directory (rather than the **examples** directory) contains the **Input.txt** and **Printout.txt** files showing the input and printout for discretizing the “Petzold average particle” phase function given on subroutine **pfpart.f**. The file containing the discretized phase function, **avgpart.dpf**, is found in the **data\phasefun** directory.

8.2 Creating a Surface Wind File

The step-by-step process is described in Section 5.2. The **surfcode** directory (rather than the **examples** directory) contains the **Input.txt** and **Printout.txt** files showing the input and printout for creating a surface wind file for a 5 m s^{-1} wind speed. The surface wind file created by this example run, **surfwind.5**, is found in the **data\surfaces** directory. This is the file used by HYDROLIGHT when a 5 m s^{-1} wind speed is selected.

8.3 A Simple Radiative Transfer Simulation

This example illustrates running HYDROLIGHT in a manner that is often used in idealized radiative transfer studies. This example solves the monochromatic RTE for a homogeneous, infinitely deep water body. For pedagogic purposes, suppose that we wish to define the inherent optical properties of the water body by the phase function and the albedo of single scattering, $\omega_o = b/c$. Here $c = a + b$ is the beam attenuation coefficient. We shall use $\omega_o = 0.75$ along with a one-term Henyey-Greenstein scattering phase function with an asymmetry parameter of $g = 0.9$. The output will be requested at various optical depths ζ , as is customary in radiative transfer studies.

This run is made by selecting the **abconst** IOP model. The input for this model needs a and b , not ω_o . However, we can use *any* values for a and b that give the desired value for ω_o , because we are requesting the output as a function of optical depth. The actual values of a and b are relevant only if we wish to convert optical depth to geometric depth via $z = \zeta/(a + b)$. Therefore, let us use $a = 0.1 \text{ m}^{-1}$ and $b = 0.3 \text{ m}^{-1}$, which give $\omega_o = 0.3/(0.1 + 0.3) = 0.75$. The desired phase function is found on file **OTHGg90.dpf**, which is one of the example discretized phase functions supplied with HYDROLIGHT. The other input for the run is as follows:

- Level sea surface (wind speed of 0)
- Idealized sky model with the sun at a 30 degree zenith angle in a uniform sky, with a diffuse-to-direct irradiance ratio of 0.3, and with a total downwelling plane irradiance $E_d = 1.0 \text{ W m}^{-2} \text{ nm}^{-1}$.
- Infinitely deep water with output at irregularly spaced optical depths (to get increased depth resolution near the surface) down to $\zeta = 30$.

The root file name used for running this example was **UGEx3** (Users' Guide Example 3). Therefore, the associated input file is **IUGEx3.txt**, the run script file is **RUGEx3.bat**, the printout is **PUGEx3.txt**, and so on. These files are all on the **examples** directory, rather than in the directories where they would normally be found.

Figure 4 shows how various K functions approach the asymptotic value K_∞ . This plot was generated by the IDL routine **ugfig4.pro**, which is found in the IDL directory. Routine **ugfig4.pro** read the file **Dugex3.txt** as input. Because this run was made in terms of optical depth, the K functions have been plotted as non-dimensional quantities: $K(\text{non-dimen}) = K(\text{dimen})/c$; see Note 3 below.

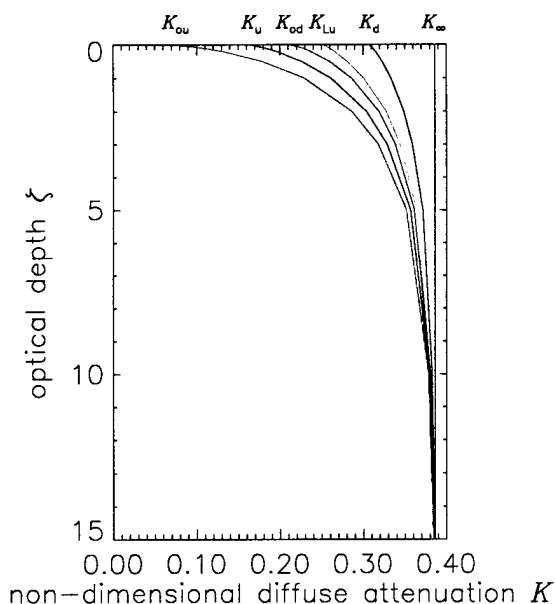


Fig. 4. Selected non-dimensional diffuse attenuation functions, as computed by the HYDROLIGHT run of Example 3. The plot is to only 15 optical depths, even though the run was to 30.

Note 1. The abconst IOP model is for runs at a single wavelength, which was taken to be 550 nm. However, the wavelength is irrelevant in this run, because the run used the idealized sky model and had infinitely deep water. Had the run used the semi-empirical sky model or had it used a finite-depth bottom with a wavelength-dependent bottom reflectance, the wavelength would have been needed. Because the run was at a single wavelength, and there was no option of including inelastic-scattering.

Note 2. For the values of $a = 0.1 \text{ m}^{-1}$ and $b = 0.3 \text{ m}^{-1}$ used in this run, 30 optical depths corresponds to $30/c = 75 \text{ m}$ of geometric depth. If you make another run with, say, $a = 1.0 \text{ m}^{-1}$ and $b = 3.0 \text{ m}^{-1}$ (and all else the same), you will find that the radiances and irradiances are all the same as a function of optical depth, even though 30 optical depths is then 7.5 m. Moreover, the run times will be the same, which is consistent with the comments in Section 7.1.

Note 3. Even though the run saved its output at the specified optical depths, the K functions in the default printout were computed using geometric-depth derivatives; the K functions therefore have units of inverse meters, as is customary. If you want non-dimensional K functions computed with optical-depth derivatives, you can multiply the values in the printout by $1/c$ ($= 2.5 \text{ m}$ in the present case). Routine `maincode\kfcn.f` which computes the K functions, also has the option of printing out a table of non-dimensional K functions.

8.4 A Simulation of Case 2 Water

As a more complicated and oceanographically relevant example of running HYDROLIGHT, consider a simulation the light field in shallow Case 2 water. The water was modeled as a four-component system:

- Component 1: pure water
- Component 2: chlorophyll-bearing particles (and covarying yellow matter)
- Component 3: CDOM (yellow matter) that does *not* covary with the chlorophyll concentration
- Component 4: mineral particles that do *not* covary with the other components.

The corresponding *ab* routine is found on file `abcase2.f`. It is emphasized that the IOP subroutine found on file `abcase2.f` is just a contrived example showing *how* to write a four-component IOP model. It is *not* intended for general use in modeling actual Case 2 waters.

The corresponding phase functions were

- Component 1: pure water (from file `pureh2o.dpf`)
- Component 2: Petzold "average particle" (from file `avgpart.dpf`)
- Component 3: irrelevant, since CDOM is non-scattering (file `pureh2o.dpf` was used)
- Component 4: Fournier-Forand with $n=1.15$, $nu = 4.0$ (from file `FF11540.dpf`)

The subroutine on file `chlzfunc.f` defines a depth-dependent chlorophyll profile. The chlorophyll concentration for this run increased from roughly 1.2 mg m^{-3} at the surface to 5.0 mg m^{-3} at $z = 5 \text{ m}$, and then decreased to 2.3 mg m^{-3} at $z = 8 \text{ m}$. The routine on file `acdom.f` defines a CDOM absorption coefficient that depends on depth and wavelength. At 440 nm, the absorption by CDOM was 0.1 m^{-1} at the surface; this value decreased exponentially to 0.02 m^{-1} at 8 m (to crudely simulate a near-surface layer of CDOM as might arise from river input). The CDOM absorption decreased exponentially with wavelength; it was negligible at red wavelengths, but was triple the above values at 350 nm.

Subroutine `abcase2` calls routines `chlzfunc` and `acdom` during the computation of the absorption and scattering coefficients at various depths and wavelengths. Standard bio-optical models (as documented in file `abcase2.f`) are used to convert the chlorophyll concentration to *a* and *b* values for the pigmented-particle component (although those models were developed for use in Case 1 waters only).

Routine `abcas2` computes the absorption and scattering coefficients for mineral particles using a Gaussian function in depth with the maximum values being at the bottom (to crudely simulate a layer of resuspended sediment) and with a negligible mineral contribution at the surface. The absorption coefficient for mineral particles had the value 0.2 m^{-1} at $z = 8 \text{ m}$ and $\lambda = 400 \text{ nm}$; the corresponding scattering coefficient was 0.8 m^{-1} . The absorption by mineral particles was assumed to depend on wavelength as λ^{-2} , and the scattering coefficient was proportional to λ^{-1} . These values for mineral particles were chosen to be comparable to those for pigmented particles, although the depth and wavelength dependencies were different for the two types of particles.

Depending on the depth and wavelength, the total absorption was dominated by water, by pigmented particles, by CDOM, or by mineral particles. Scattering by particles was always much greater than scattering by the water; the CDOM was assumed to be non-scattering. This model for a and b gave albedos of single scattering, ω_0 , that ranged from less than 0.3 (near the surface, at 700 nm, where absorption by the water dominated) to more than 0.85 (near 570 nm, where absorption was lowest). The geometric depth of 8 m corresponded to over 13 optical depths at 350 nm (where both CDOM absorption and particle scattering were high) and to less than 8 optical depths near 570 nm.

The run included Raman scattering by the water itself and fluorescence both by chlorophyll and CDOM. The simulation therefore included wavelengths from 350 to 700 nm, as is necessary when fluorescence effects are included. Wavelength bands were 10 nm wide between 350 and 670 nm and 5 nm between 670 and 700 nm. These wavelength bands were chosen to approximate the bandwidths typically used by hyperspectral remote sensing instruments and to give higher resolution in the chlorophyll emission region near 685 nm. There was no bioluminescence in the run.

The bottom was placed at a depth of 8 m. Output was requested every half meter between the surface and the bottom, for a total of 17 depths. The bottom was modeled as a Lambertian surface with a wavelength-dependent irradiance reflectance based on measurements of a green algae; the bottom reflectance data are on file `data\botmref\Nrgreenal.bot`.

The "semi-empirical" sky model was used with a "Julian" day of 166, latitude of 48.5 degrees North, longitude of 123 degrees West, and Greenwich mean time of 23 hours (which correspond to mid-afternoon on June 15 in Puget Sound, Washington, USA). The corresponding solar zenith angle was 41.55° . The solar azimuthal angle relative to the downwind direction was (by default) 0° . The sky was assumed to be clear. The wind speed was taken to be 5 m s^{-1} .

This run required less than 30 minutes on a 400 MHz Pentium Processor. It should be noted that the simulation required the solution of the radiative transfer equation for 38 wavelength bands, with three types of inelastic scattering connecting the bands. On average, each wavelength solution was taken to about ten optical depths. Thus the run time corresponds to about five seconds per wavelength per optical depth. The archival printout file `Pcase2.txt` is less than 1 Mbyte, but the digital output file `Dcase2.txt` (which contains the radiance distribution at all depths, directions, and wavelengths) is about 13 Mbytes.

The various input and output files associated with this simulation are found on the `examples` directory. The following figures show some of the input to and output from this simulation. The IDL plot routines that generated these figures are found on the `IDL` directory.

Figure 5 shows the IOP's a , b , and ω_0 as functions of depth and wavelength. (This figure was generated using IDL routine `ugfig5.pro`.)

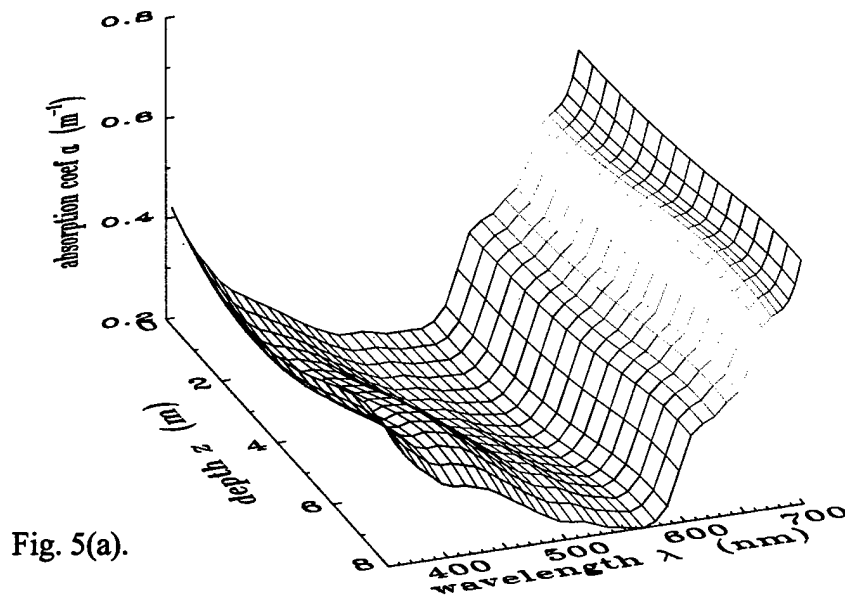


Fig. 5. Inherent optical properties as function of depth and wavelength, as used in Example 4. Panel (a) is the absorption coefficient a , panel (b) is the scattering coefficient b , and panel (c) is the albedo of single scattering ω_0 .

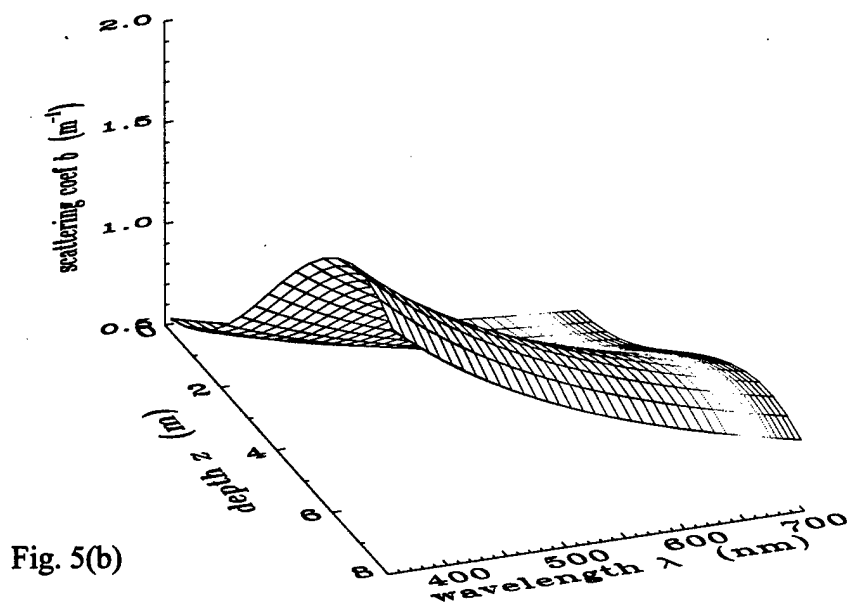


Fig. 5(b)

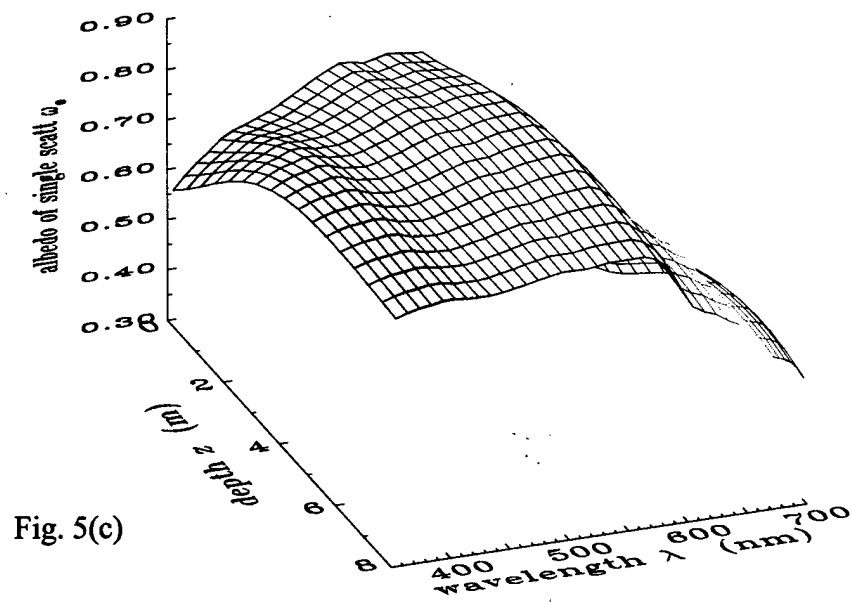


Fig. 5(c)

The graphical information of Fig. 5 can be understood quantitatively by examination of the default printout on file **Pcase2.txt**. The printout shows, for example, that the total absorption is high near the surface at UV wavelengths (λ near 350 nm) because of the high absorption by CDOM; the total is high near the bottom at UV wavelengths because of absorption by the mineral particles; the total is high at red wavelengths because of absorption by water itself, and so on. The total scattering in the UV is high at mid-depths because of the pigmented particles; it remains high near the bottom because of the combined effects of pigmented particles and mineral particles; and so on.

Figure 6 shows how the radiance distribution in the azimuthal plane of the sun changes with depth, polar angle, and wavelength. (This figure was generated using IDL routine **ugfig6.pro**.)

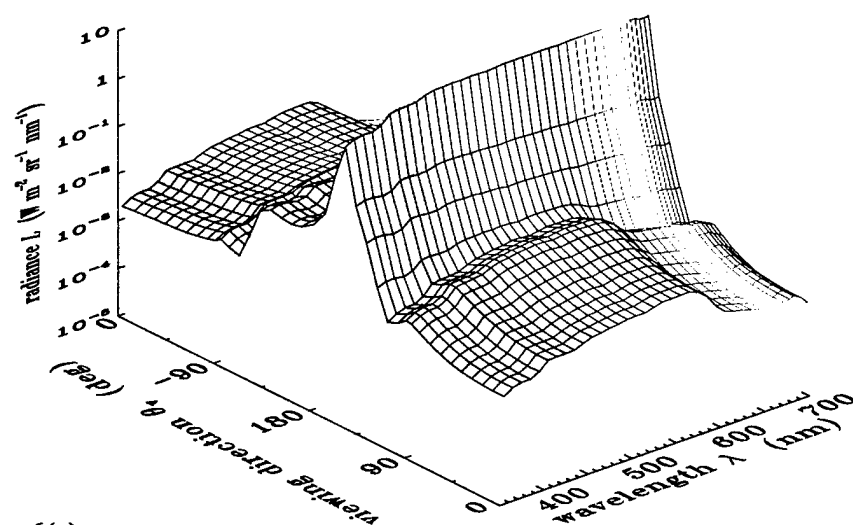


Fig. 6(a)

Fig. 6. The radiance distribution in the plane of the sun's rays, as a function of viewing direction and wavelength. Panel (a) is a $z = 0$, just beneath the air-water surface; Panel (b) is at the mid-water depth of 4 m; Panel (c) is at the bottom, $z = 8$ m.

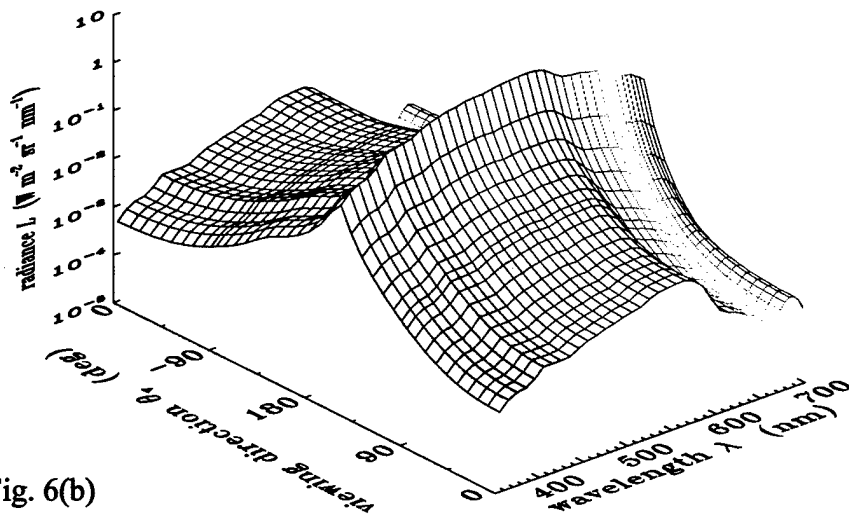


Fig. 6(b)

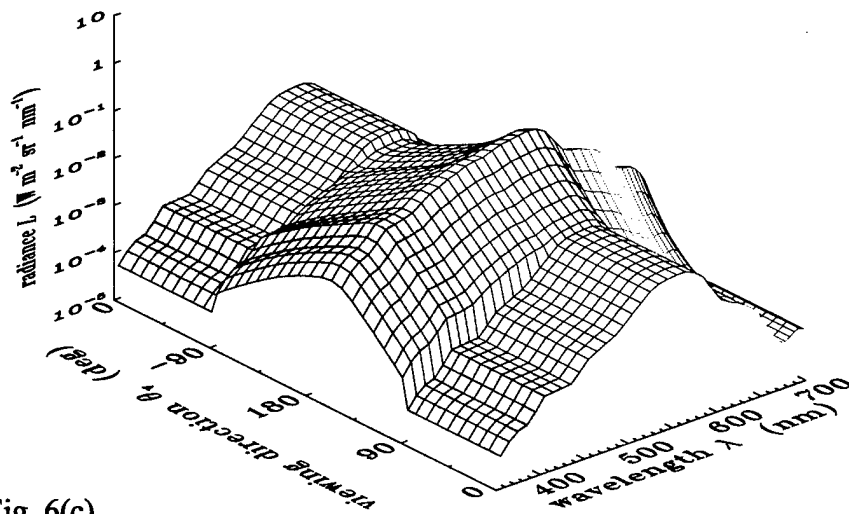


Fig. 6(c)

The polar angle in Fig. 6 is the viewing direction, i.e., the direction a sensor points in order to detect photons traveling in the opposite direction. Because HYDROLIGHT measures the polar angle from 0 at the nadir, a viewing direction of $\theta_v = 180^\circ$ is looking straight up, seeing the radiance heading straight down; $\theta_v = 0$ corresponds to looking straight down and seeing the upwelling radiance L_u . In this figure, positive θ_v values represent azimuthal directions looking toward the sun, and negative θ_v values represent azimuthal angles looking away from the sun. The radiance maxima at $\theta_v \approx 30^\circ$ (i.e., looking toward the sun) are due to the sun's direct rays after refraction at the air-water surface. These maxima become less prominent with depth as scattering makes the radiance distribution more diffuse. It should be noted in Fig. 6(c) that at the bottom, $z = 8$ m, the upwelling radiance (viewing angles between -90° and $+90^\circ$) is constant with polar angle. This is a consequence of modeling the bottom as a Lambertian surface. The “bumps” in the upwelling radiance near 685 nm are a consequence of chlorophyll fluorescence.

The Dcase2.txt file of course contains the complete radiance distribution at all depths, wavelengths, and directions. It is therefore possible to compute any desired radiometric or apparent optical property from the information on this file. For example, Fig. 7 shows the diffuse attenuation function for upwelling radiance, K_{Lu} , as a function of depth for selected wavelengths. Note that K_{Lu} becomes negative near the bottom at 535 nm, i.e., the upwelling radiance increases with depth. This is a consequence of the high bottom reflectance at green wavelengths. This increase in L_u near the bottom can be seen by comparing Figs. 6(b) and 6(c) for $\theta_v = 0$ and green wavelengths.

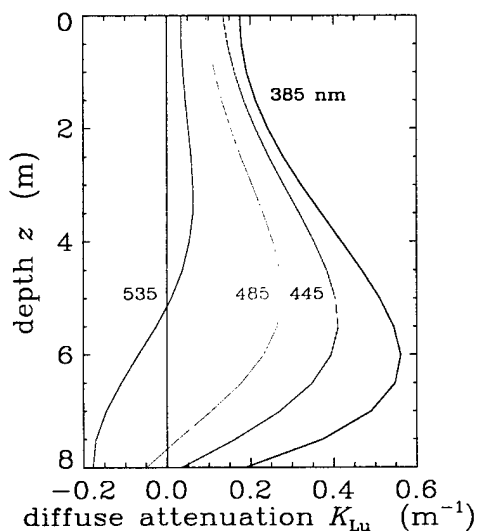


Fig. 7. Diffuse attenuation for upwelling radiance at selected wavelengths. (This figure was generate by IDL routine ufig7.pro.)

As a final example of the type of output than can be obtained from HYDROLIGHT, Fig. 8 shows the remote-sensing reflectance $R_{rs} = L_w/E_d$ (where L_w is the water-leaving radiance, i.e., the total radiance minus the reflected sky radiance; and E_d is evaluated just above the water surface). Note that HYDROLIGHT computes the water-leaving and reflected-sky radiances separately, so the remote-sensing reflectance show here is exact, not an approximation obtained via any semi- empirical correction for reflected-sky radiance or for transmitted upwelling radiance from below the surface. As expected for this “green” Case 2 water, R_{rs} is greatest at green wavelengths. The bump at 685 nm shows the effect of chlorophyll fluorescence.

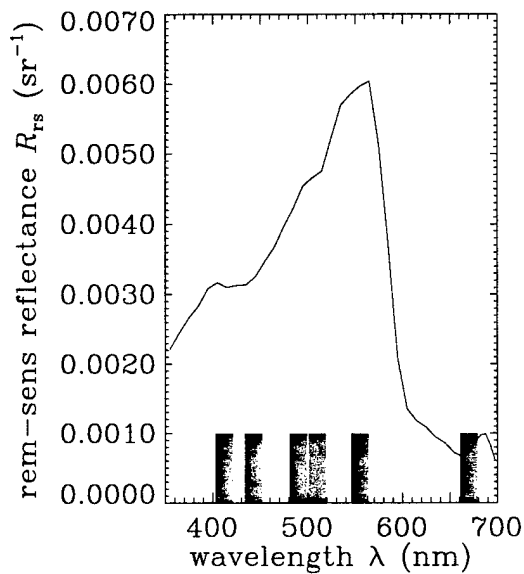


Fig. 8. The remote-sensing reflectance as a function of wavelength. The bars at the bottom of the figure show the nominal SeaWiFS sensor bands. (This figure was generated by IDL routine ugfig8.pro.)

REFERENCES

- Dongarra, J. J. and E. Grosse, 1987. Distribution of mathematical software via electronic mail, *Commun. ACM*, 30(5), 403-407.
- Gregg, W. W. and K. L. Carder, 1990. A simple spectral solar irradiance model for cloudless maritime atmospheres, *Limnol. Oceanogr.*, 35(8), 1657-1675.
- Harrison, A. W. and C. A. Coombes, 1988. An opaque cloud cover model of sky short wavelength radiance, *Solar Energy*, 41(4), 387-392.
- Kirk, J. T. O., 1994. *Light and Photosynthesis in Aquatic Ecosystems*, Cambridge University Press, Cambridge, 509 pages.
- Kneizys, F. X., E. P. Shettle, L. W. Abreu, J. H. Chetwynd, G. P. Anderson, W. O. Gallery, J. E. A. Selby, and S. A. Clough, 1988. Users guide to LOWTRAN 7, Air Force Geophysics Lab, Rept AF6L-TR-88-0177, Hanscom AFB, MA 137 pages.
- L&W, see Mobley (1994)
- Light and Water, see Mobley (1994)
- LOWTRAN, see Kneizys, et al. (1988).
- Mobley, C. D., 1994. *Light and Water: Radiative Transfer in Natural Waters*, Academic Press, San Diego, 592 pp.
- Mobley, C. D., 1995. *The Optical Properties of Water*, Chapter 43 in *Handbook of Optics, Second Edition, Volume I*, M. Bass, Editor in Chief, McGraw-Hill and Optical Society of America, New York, 56 pages.
- Mobley, C. D., B. Gentili, H. R. Gordon, Z. Jin, G. W. Kattawar, A. Morel, P. Reinersman, K. Stamnes, and R. H. Stavn, 1993. Comparison of numerical models for computing underwater light fields, *Appl. Opt.*, 32, 7484-7504.

Spinrad, R. W., K. L. Carder, and M. J. Perry, 1994. *Ocean Optics*, Oxford University Press, New York, 283 pages.

APPENDIX A. DETAILED DESCRIPTION OF THE RUN-TIME INPUT FOR STANDARD RUNS

The various subroutines and data files used by HYDROLIGHT provide much of the information needed to make a standard run. The remaining information is read in at run time from the `lroot.txt` file. This file is generated automatically by the front-end programs, and its exact format is not of interest to most users. However, some users may wish to edit this file to change the input from one run to the next, rather than running a front-end program. Advanced users may wish to alter the input in order to tailor HYDROLIGHT to their specific needs. This can be done by changing subroutine `maincode\initial.f`, which reads the `lroot.txt` file. The following pages describe in detail the input records found on file `lroot.txt`. Each of the records is free format. The names of the variables are those used in `initial.f`, which reads `lroot.txt`; additional documentation is given in `initial.f`.

RECORD 1.

This record gives the descriptive title for the run. The title can be up to 120 characters long.

Name of variable: *ititle*

Example: Users' Guide, Example 1

RECORD 2.

This record gives the root name to be used for file generation.

Name of variable: *rootname*

Example: `ugex1`

RECORD GROUP 3.

The first record gives the number of components expected in the IOP model (the number of components built into the "ab" routine):

Name of variable: *ncomp*

Example: 3

The next *ncomp* number of records give the names of the files containing the discretized phase

functions to be used with each component of the IOP model. The order MUST match the order of the components in the subroutine for *a* and *b*; see Section 6.1.

Names of variables: *pfname*(1) to
pfname(*ncomp*)

Example: pureh2o.dpf
avgpart.dpf
othgg90.dpf

RECORD GROUP 4.

These records define the wavelengths to be used in the run. The first record gives *nwave*, the number of wavelength BANDS at which the model is being run:

Name of variable: *nwave*

Example: 5

The format of the next record is determined by the value of *nwave*:

if *nwave* = 0, the run is to be made at an exact wavelength. In this case, the next record gives

Names of variables: *wavel*, *areset*, *breset*

Example: 532.0, 0.1, 0.25

where *wavel* is the EXACT WAVELENGTH in nm, and *areset* and *breset* are the values of *a* and *b* to be used in the *abconst* IOP model, if it is being used. Otherwise, *areset* and *breset* have values of -1.0. The sky spectral radiance at the exact wavelength *wavel* will be used (1 nm resolution).

if *nwave* ≥ 1, the run is to be made with one or more finite wavelength bands. In this case, the next record gives

Names of variables: *waveb*(1), *waveb*(2), ..., *waveb*(*nwave*+1)

Examples: 400.0, 420.0, 430.0, 440.0, 450.0, 475.0

where the values of *waveb*(*j*) give the *nwave*+1 WAVELENGTH BAND BOUNDARIES (in nm) for which the model is to be run. (More than one record can be used if needed to list all of the band boundaries.) *a* and *b* values as returned by the IOP model at the band centers will be used. The band-averaged sky radiance will be

used.

RECORD 5.

Gives the flags specifying whether or not internal sources and inelastic scatter are to be included in the run.

Names of variables: *ibiolum*, *ichlfl*, *icdomfl*, *iraman*

Example: 0, 1, 0, 1

ibiolum is a flag for the inclusion/omission of bioluminescence:

if *ibiolum* = 0, there is no bioluminescence present.

if *ibiolum* = 1, the run includes bioluminescence; user-supplied routine *s0biolum* is required.

ichlfl is a flag for the inclusion/omission of chlorophyll fluorescence:

if *ichlfl* = 0, there is no chlorophyll fluorescence present.

if *ichlfl* = 1, chlorophyll fluorescence is present; routine *chlzfunc* or *chlzdata* is called.

icdomfl is a flag for the inclusion/omission of CDOM fluorescence:

if *icdomfl* = 0, there is no CDOM fluorescence present.

if *icdomfl* = 1, CDOM fluorescence is present; routine *acdom* is required.

iraman is a flag for the inclusion/omission of Raman scattering:

if *iraman* = 0, there is no Raman scattering present.

if *iraman* = 1, Raman scattering is present.

RECORD 6.

This record gives information needed by whichever sky radiance model is being used in the run. The general form of the record is

Names of variables: *iflagsky*, *nsky*, *skydata(1)*, *skydata(2)*, ..., *skydata(nsky)*

iflagsky = 1 if the "idealized" sky models are being used

= 2 if the "semi-analytic" sky model is being used, with solar zenith angle being specified

= 3 if the "semi-analytic" sky model is being used, with time and location being specified

nsky is the number of values to be read in the remainder of the record.

The format of the record depends on which sky model is being used:

If *iflagsky* = 1, then *nsky* = 5 and *skydata*(1) to *skydata*(5) contain

Names of variables: *suntheta*, *sunphi*, *C*, *rsky*, *Edtotal*

Example: 30.0, 0.0, 1.25, 0.3, 1.2

where

suntheta is the solar zenith angle in degrees; *suntheta* = 0.0 for the sun at the zenith and *suntheta* = 90.0 for the sun at the horizon.

sunphi is the solar azimuthal angle in degrees *relative to the wind direction*. *sunphi* = 0.0 is downwind and *sunphi* = 90.0 places the sun at a right angle to the wind. The default is *sunphi* = 0.0.

C is the cardioidal parameter *C* in *Light and Water* Eq. (4.50)]. *C* = 0.0 gives a uniform background sky and *C* = 2.0 gives a cardioidal sky; *C* = 1.0 gives a heavy overcast sky.

rsky is the ratio of background-sky to total scalar irradiance, $0.0 \leq rsky \leq 1.0$. *rsky* = 0.0 for a black sky (sun only) and *rsky* = 1.0 for a fully overcast sky (no sun visible).

Edtotal is the downwelling spectral scalar irradiance due to sun and background sky incident onto the sea surface; the units are $\text{W m}^{-2} \text{nm}^{-1}$.

If *iflagsky* = 2, then *nsky* = 3 and *skydata*(1) to *skydata*(3) contain

Names of variables: *thetas*, *phis*, *cloud*

Example: 45.0, 0.0, 0.0

where

suntheta is the solar zenith angle, defined as above.

sunphi is the solar azimuthal angle, defined as above.

cloud is the cloud cover, $0.0 \leq cloud \leq 1.0$. *cloud* = 0.0 for a clear sky and *cloud* = 1.0 for a solid overcast.

If *iflagsky* = 3, then *nsky* = 6 and *skydata*(1) to *skydata*(6) contain

Names of variables: *ffday*, *rlat*, *rlon*, *hour*, *phis*, *cloud*

Example: 186.0, 30.0, -125.255, 20.5, 0.0, 0.5

ffday is the Julian day: *ffday* = 1.0 for January 1.

rlat is the latitude in degrees; positive for north and negative for south.

rlon is the longitude in degrees; positive for east and negative for west.

hour is Greenwich Mean Time in hours (Pacific Standard Time plus 8 hours). Minutes are expressed as a fraction, e.g., 21.40 is 9:25 PM GMT.

phis is the azimuthal angle, defined as above.

cloud is the cloud cover, defined as above.

Other parameters required for the sky radiance computations are set to default values (see Section 4.4), but could be read in here if desired after minor modification of *maincode\initial.f*.

Note 1. If you wish to write your own sky model, you can modify the front-end programs to read in the data needed by your sky model using the *skydata* array. Then pass the needed data on to your model in the manner seen in routine *initial.f*.

Note 2. The general Cox-Munk capillary wave slope statistics used to model the sea surface depend on the angle relative to the wind direction. Add more here.....

RECORD 7.

This record gives the wind speed in meters per second at an anemometer height of 12 m.

Name of variable: *iwindspd*

Example: 5

The integer value of *iwindspd* is used to generate the file extension needed to locate the proper file of surface data in the *data\surfaces* directory.

RECORD 8.

This record defines the type of bottom boundary.

Names of variables: *ibotm*, *rflbot*

Example: 1, 0.2

ibotm is a flag for the type of bottom boundary, as follows:

if *ibotm* = 0, the water column is infinitely deep. The water below depth z_{\max} (to be specified in record __ below) is taken to be homogeneous with IOP's equal to the values at depth z_{\max} . The bi-directional radiance reflectance of the infinite layer of water below depth z_{\max} is computed automatically from the IOP's.

if *ibotm* = 1, the bottom is an opaque Lambertian reflecting surface located at depth z_{\max} . The irradiance reflectance of the bottom is taken to be *rflbot*, independent of wavelength. Note that $0 \leq \text{rflbot} \leq 1$.

if *ibotm* = 2, the bottom is an opaque Lambertian reflecting surface located at depth z_{\max} . The wavelength-dependent irradiance reflectance of the bottom will be read from a HYDROLIGHT standard format file of bottom reflectance data (see Section 6.7).

A value of *rflbot* is always read, but is used only if *ibotm* = 1.

RECORD 9.

This record gives the depths at which output is to be saved for post-run analysis. The depths as read can be either dimensionless optical depths ζ or geometric depths z in meters. The last depth specified is taken to be the "maximum depth of interest," z_{\max} , where the bottom boundary condition will be applied. (The water below the last depth is assumed to be homogeneous if *ibotm* = 0. If *ibotm* \geq 1, the Lambertian bottom is placed at the last output depth.) The record has the form

Names of variables: *iop*, *nznom*, *zetanom*(1), *zetanom*(2), ..., *zetanom*(*nznom*)

Example: 0, 5, 0.0, 1.0, 2.0, 5.0, 10.0

iop is a flag for optical or geometric depth, as follows:

if *iop* = 0 then the *zetanom* values are GEOMETRIC depths in meters.

if *iop* = 1, then the *zetanom* values are OPTICAL depths.

$nznom$ is the number of depths where output is desired.

$zetanom(1) = 0.0, \dots, zetanom(nznom) = z_{\max}$ are the depths where output is desired.

Note that if the run contains inelastic scattering, then the run *must* use geometric depth. Note also that $zetanom(1)$ must always be 0.0, the depth in the water just below the mean air-water surface, and that $zetanom(nznom)$ is by definition the maximum depth of interest. Recall from the discussion of Section 4.2 that a small increment will be added to each nominal output depth for the purpose of computing depth derivatives for K functions.

RECORD 10.

This record gives the name of the file of ac-9 data to be read by “*ab*” routine *abac9*. If no ac-9 data file is needed (i.e., if “*ab*” routine *abac9* is not being used in the run), a dummy file name is written.

Name of variable: *ac9datafile*

Example: *myac9data.txt*

RECORD 11.

This record gives the name of the file of chlorophyll data to be read by routine *chlzdata*. If no chlorophyll data file is needed (i.e., if routine *chlzdata* is not being used in the run), a dummy file name is written.

Name of variable: *chlzdatafile*

Example: *mychldata.txt*

RECORD 12.

This record gives the name of the file of bottom reflectance data to be read by routine *rbottom*. If no bottom-reflectance data file is needed (i.e., if *ibotm* = 0 or 1), a dummy file name is written.

Name of variable: *rbottomdatafile*

Example: *Rcoralsd.bot*

APPENDIX B. INSTALLING THE HYDROLIGHT CODE

This APPENDIX gives the details of how to install the HYDROLIGHT code on your computer. Installing and running the code is different for WINDOWS and UNIX systems.

B.1 Computer Requirements

The HYDROLIGHT 4.0 source code is written entirely in FORTRAN 77 in order to make the code as portable as possible. Previous versions of HYDROLIGHT have been run on a variety of computers including CDC and Cray mainframes; Sun, Silicon Graphics, and DEC UNIX workstations; IBM and IBM-clone Personal Computers (PCs) with 486 and Pentium processors running Microsoft Windows 3.1 and Windows 95, and Apple Power Macintoshes. Version 4.0 should be equally portable to any machine with a FORTRAN 77 or FORTRAN 90 compiler.

Although the HYDROLIGHT code can run on a wide range of computers, most users will prefer to run Version 4.0 on a PC with the Microsoft Windows 95, 98, or NT operating system. *A PC/Windows platform for HYDROLIGHT is strongly recommended* for several reasons:

- The HYDROLIGHT software system (optionally) includes a subset of the Lahey FORTRAN 90 version 4.5 compiler, which is designed for Pentium (or compatible) processors and Microsoft Windows operating systems.
- The convenient GUI front end runs only on Windows 95/98/NT.
- Excel spreadsheets software is usually found on PCs with Windows 95/98/NT.
- Related software, such as the IDL graphics package, is much cheaper for PCs than for mainframes or UNIX workstations.

The HYDROLIGHT package of source code and data files requires about 20 Mbytes of storage as distributed. The Lahey compiler, as optionally distributed with HYDROLIGHT, requires an additional 20 Mbytes. After compilation on the user's computer, the executable file for the main code is less than 3 Mbytes. The minimum required amount of random access memory (RAM) is therefore not large. However, compilation (and perhaps also run) times can be very slow if insufficient RAM forces extensive swapping. Experience shows that 30 Mbytes of RAM is more than enough for satisfactory compilation and running of Hydrolight. The Droot.txt output files can be 10 Mbytes or larger for runs requesting output at many depths and wavelengths. Thus, disk storage can be consumed quickly in a series of

simulations requesting extensive output (e.g. for high-resolution graphics). Moreover, some of the internal scratch files used during the solution of the radiative transfer equation can be many tens of Mbytes if many depths and wavelengths are included in the run. Thus 100 Mbytes of free disk space is recommended for running HYDROLIGHT.

The GUI front-end program is designed for optimum use on monitors with 1024 x 768 pixels. Lower-resolution monitors cause some of the GUI forms to display as larger than the monitor screen. Higher-resolution monitors make the GUI forms appear smaller than intended, which makes the forms harder to read.

The front-end programs issue commands to invoke the compiler and linker. It is generally necessary to recompile a few subroutines for each run, because the include files (containing information on which IOP and sky models are being used, etc.) are updated by the front end program, and consequently the HYDROLIGHT core routines containing these include files must be recompiled to guarantee that all files are up to date. The commands necessary for invoking the compiler are different for different brands of compilers, and even for different versions of the same compiler. In order to avoid the considerable expense required in supporting various compilers, Sequoia Scientific, Inc. has a licensing agreement with Lahey Computer Systems, Inc. that allows the distribution of particular files of the Lahey FORTRAN 90 version 4.5 compiler with HYDROLIGHT. These are the files needed to compile and link the HYDROLIGHT code. The Lahey files distributed with HYDROLIGHT do not contain the features of the complete Lahey FORTRAN 90 software system. *Because the necessary part of the Lahey compiler is available (at the licensee's request) with HYDROLIGHT, no support is given for other FORTRAN compilers when HYDROLIGHT is run on a PC. Appendix C gives important conditions regarding the user of the Lahey compiler.*

UNIX "makefiles" are provided with HYDROLIGHT for accomplishing the compilation and linking tasks under the Sun Microsystems, Inc. Solaris® 2.5.1 UNIX operating system the Sun Microsystems FORTRAN 77 version 4.0 compiler. Other UNIX systems can usually use the same makefiles, but *support is not provided for other UNIX systems or compilers.*

B.2 Installation on Microsoft Windows Operating Systems

This section describes how to install HYDROLIGHT on a PC running the Microsoft Windows 95/98/NT operating system. It is assumed that you will be running HYDROLIGHT with the Lahey FORTRAN 90 version 4.5 compiler that comes with HYDROLIGHT.

The HYDROLIGHT software package is distributed as one (uncompressed) directory named H40 (with subdirectories as seen in Fig. 3) on a CD or Iomega ZIP disk. The description below assumes you are installing HYDROLIGHT from a CD to the C drive on your computer. The procedure is the same if you are installing from a ZIP disk or to another drive. To install the code on your PC, do the following:

Step 1. Insert the CD into its drive.

Step 2. Use Windows Explorer to drag the entire H40 directory from the CD to your C drive. (If you wish to rename the main HYDROLIGHT directory from H40 to something else, you can do this as follows: open Windows Explorer, right click on H40, select RENAME and then enter the new name for the directory. If you do this, use the new directory name in all of the steps below.)

Step 3. Place a shortcut with the HYDROLIGHT icon on your desktop, using the following steps:

- Open Windows Explorer and go to the c:\H40\frontend directory
- Right click on the h40winfe.exe file and select CREATE SHORTCUT
- Drag shortcut to h40winfe.exe to the desktop
- Right click on the shortcut.... icon and select RENAME. Change the name shortcut... to whatever you want, e.g. H40 or Hydrolight
- Right click again on the desktop icon and select PROPERTIES → SHORTCUT → CHANGE ICON → BROWSE. Go to directory c:\h40\frontend and select the H40icon.ico file. Then click OPEN. The “change icon” window should now show the HYDROLIGHT icon under “current icon”. Back out by clicking OK twice and close Windows Explorer. The HYDROLIGHT icon should now be on the desk top.

Step 4. When the HYDROLIGHT 4.0 files are written to the CD, the software that creates the CD automatically sets each file to “read only” status (because the CD is a read-only medium). However, some files (such as the include files in the maincode directory and the HFEdfits.txt and HFEreset.txt files in the frontend directory) must be overwritten each time HYDROLIGHT runs. Other files are overwritten if you make changes in the source code. It

is therefore necessary to remove the read-only flags from at least some of the files after they are copied to your computer. The H40 directory contains a batch file named H40setup.bat

that will remove the read-only flag from every file in all of the H40 subdirectories. You can run this batch file as follows:

- Open a Command (DOS) window and go to the H40 directory. Enter the command

```
H40setup.bat
```

It is not necessary to run this batch file if you have installed HYDROLIGHT from a Zip disk, because the original flag settings are not changed when a Zip disk is created.

If you already have installed your own copy of Lahey FORTRAN, the installation of HYDROLIGHT is now complete. The Lahey install program will have modified your autoexec.bat file to add the path where Lahey FORTRAN was installed (the default is C:\LF9045\BIN for version 4.5). If you are going to use the subset of Lahey FORTRAN version 4.5 that (optionally) comes with HYDROLIGHT, you must manually modify your autoexec.bat file so that it has the search path needed to find the Lahey compiler distributed with HYDROLIGHT. Do this as follows:

Step 5. Add the search path for Lahey FORTRAN to the autoexec.bat file, using the following steps:

- Make a backup of your present c:\autoexec.bat file. One way to do this is to open a DOS window (e.g., via START → PROGRAMS → MS-DOS PROMPT), go to the C drive, and enter the DOS command

```
copy autoexec.exe autoexec.old
```

- Open the autoexec.bat file with an ASCII text editor (such as notepad, via START → PROGRAMS → ACCESSORIES → NOTEPAD).
- Add the following lines *exactly as shown* (no extra spaces; start at the left margin) to the *end* of the autoexec.bat file:

```
REM - Prepend PATH variable - for Lahey Fortran 90 v4.5
REM - as distributed with HYDROLIGHT 4.0
SET PATH=C:\H40\LF9045\BIN;"%PATH%"
```

- Re-save the modified **autoexec.bat** file. Make sure you save the file as a TEXT DOCUMENT.
- Restart your computer, so that the new **autoexec.bat** file will take effect.

Some recent versions of the Microsoft 95/98/NT operating systems do not come with an **autoexec.bat** file, although such a file can still be used. If your computer does not contain a file name **autoexec.bat**, you can create one with an ASCII text editor. Just create a file with the lines (starting at the left margin, column 1)

```
@ECHO OFF
PATH C:\H40\LF9045\BIN
```

Save the file in the **C** directory with the name **autoexec.bat** and reboot the computer. Then check the path as described in Note 3 below to see if the path includes the Lahey Fortran directory.

HYDROLIGHT is now installed. You can start a HYDROLIGHT run by double clicking on the HYDROLIGHT icon on the desktop.

Note 1. If double clicking on the HYDROLIGHT icon gives an error message about not being able to find the target file, right click on the HYDROLIGHT icon, select **PROPERTIES** → **SHORTCUT** and make sure the target is **C:\H40\frontend\H40winfe.exe**.

Note 2. If the GUI gives the error message "Error 75: File/path access denied" when you click **CONTINUE** on the **OUTPUT DEPTHS** form the first time you run **HYDROLIGHT**, this probably means that certain files still have the read-only attribute, as described in Step 4 above. You can check as follows to see if this is the case:

- Open a Command (DOS) window and go to the **maincode** directory. Enter the DOS "attribute" command

`attrib`

to display the attributes of all the files in the `maincode` directory. You will probably see something like

```
R      FILENAME      FULL FILE PATH
```

for each file. The “R” is the flag for read only.

- Remove the read-only attribute by performing Step 4 above, or by entering the DOS command

```
attrib -r +a *
```

The “-r” will remove the read-only attribute and the “+a” will set the “archive” attribute, which says the file has been changed since it was last backed up (on your computer). The “*” says do this for every file in the directory.

Note 3. If the GUI front end runs to completion, spawns a command window, but then fails to find the Lahey compiler, open a command window and enter the DOS command `path` to see the current search path. You should see something like

```
PATH=C:\H40\LF9045\BIN;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\...
```

If the PATH does not contain a path for the Lahey compiler, e.g., `C:\H40\LF9045\BIN`, then the path was not properly added to the `autoexec.bat` file or the computer was not rebooted after making the addition.

B.3 Installation on UNIX Systems

The code for UNIX systems is contained on a single uncompressed “tar” file named `H40.tar` on the CD or Zip disk. Install HYDROLIGHT on a UNIX system as follows:

- Copy the H40.tar file to the directory where you want to install HYDROLIGHT (e.g., /usr/home)
- Enter the UNIX command

```
tar xvf H40.tar
```

to extract all of the files from the tar file. This will place the entire H40 directory structure on your computer (e.g., as /usr/home/H40).

Note that the UNIX version of the H40 directory does not contain the LF9045 or guicode directories, which are applicable only to Windows systems. For the same reason, various files (such as the GUI front end executable file and the Lahey automake.* files) are omitted from various directories.

When running HYDROLIGHT on a UNIX workstation, the text-based front end program found on file H40txtFE.f (HYDROLIGHT 4.0 text front end) in the frontend directory can be used. This program and the subroutines it calls are found in the frontend directory. You can run this front-end program by compiling these Fortran routines and running the executable file from a command window. Both the text-based and GUI front-end programs read and write exactly the same files.

The maincode, surfcode, discpf, and frontend directories all contain UNIX makefiles for compiling and linking the HYDROLIGHT code in these directories. These makefiles are always named makeunix. To compile the text based front-end program, for example, go to the frontend directory and enter the UNIX command

```
make -f makeunix
```

from a command line prompt. This will compile the code for the text-based front end program and create an executable file named H40txtFE.exe (which can be renamed for convenience). The text-based front-end program is then run by entering the command

```
H40txtFE.exe
```

from a command line prompt.

B.4 The Lahey Automake Configuration File Editor

The first time the Lahey compiler runs in a given directory, it examines all of the files in the directory to determine which files must be compiled and to determine the interdependencies of the various files. This examination is done by the automake utility, and the results are written to several files named `automake.*` in the directory. On subsequent runs, the `automake.*` files contain the information needed for compiling and linking. If the `automake.*` files are not already available, the following window will appear:

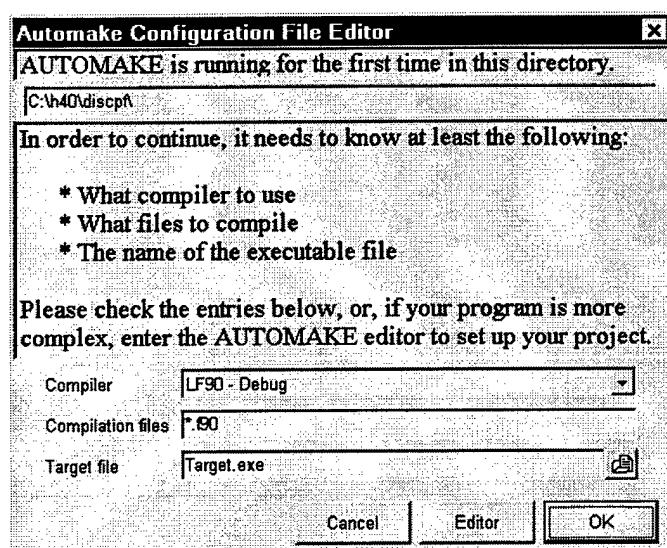


Figure 9. The Lahey Automake Configuration File Editor window.

The following responses give automake the information it needs for running HYDROLIGHT :

Compiler:	default: LF90 - Debug	leave as is
Compilation files:	default: *.f90	change to *.f
Target file:	default: Target.exe	change to maincode.exe if you are in the maincode directory; change to surfcode.exe if you are in the surfcode directory; change to discpf.exe if you are in the discpf directory

APPENDIX C. LICENSE AGREEMENT FOR USE OF THE LAHEY FORTRAN COMPILER AS DISTRIBUTED WITH HYDROLIGHT 4.0

Licensees of HYDROLIGHT 4.0 may, at their request, receive a subset of the Lahey Computer Systems, Inc. Fortran 90 Version 4.5 compiler. This compiler is designed for use on Pentium or compatible processors running the Microsoft Windows 95, 98, or NT operating systems. The subset of the compiler as optionally distributed with HYDROLIGHT 4.0 contains only the software needed to compile, link, and run HYDROLIGHT. The subset does not contain the various development tools, debuggers, and other features of the full Lahey Fortran 90 Version 4.5 software package. Sequoia Scientific, Inc. has licensed the right as an Original Equipment Manufacturer to distribute a subset of the full compiler with HYDROLIGHT 4.0. The conditions under which users who receive this subset of the Lahey Computer Systems, Inc. FORTRAN 90 Version 4.5 compiler can use that compiler are as follows:

- (1) The Lahey software is licensed for use ONLY on the single computer where HYDROLIGHT 4.0 is licensed for use. Under no circumstances is the Lahey software to be installed on more than one computer at a time or given to any third party.
- (2) Users of the Lahey software that comes with HYDROLIGHT 4.0 are NOT entitled to receive any user support, documentation, or other assistance from Lahey Computer Systems, Inc. and must not contact Lahey Computer Systems, Inc. requesting such support or documentation. Any questions or problems related to the use of the Lahey software when running HYDROLIGHT 4.0 must be referred to Curtis D. Mobley at Sequoia Scientific, Inc.
- (3) Curtis D. Mobley and Sequoia Scientific, Inc. will provide user support for the use of the Lahey software only when it is used to run HYDROLIGHT 4.0, and that user support may be limited by the support given to Sequoia Scientific, Inc. by Lahey Computer Systems, Inc. Sequoia Scientific, Inc. is not responsible for any "bugs" or other problems associated with the Lahey software itself or with its failure to operate properly on the user's computer.
- (4) Neither Lahey Computer Systems, Inc., Sequoia Scientific, Inc., nor Curtis D. Mobley shall be liable for any lost or anticipated profits or any other damage resulting from the use of

the Lahey software. Neither Lahey Computer Systems, Inc., Sequoia Scientific, Inc., nor Curtis D. Mobley shall be liable for any incidental, exemplary, special, or consequential damages, regardless of whether Lahey Computer Systems, Inc., Sequoia Scientific, Inc. or Curtis D. Mobley was advised of the possibility of such damages.

Users who wish to obtain the full Lahey Fortran 90 Version 4.5 software package, including documentation and user support, must license the full software package from Lahey Computer Systems, Inc.

APPENDIX D. LICENSE AGREEMENT FOR USE OF HYDROLIGHT 4.0

This license agreement is also contained in the source code and in the GUI front end.

HYDROLIGHT LICENSE AGREEMENT

HYDROLIGHT is Copyright (c) 1992-1998 by Curtis D. Mobley. HYDROLIGHT is owned by Curtis D. Mobley and is protected by United States copyright laws and international treaties.

This computer program, named HYDROLIGHT and consisting of various main programs and subroutines hereafter referred to collectively as "HYDROLIGHT" is being LICENSED (NOT SOLD) to the User on a non-exclusive, non-transferable basis for use in scientific research.

The following requirements, which with the preceding paragraphs constitute an agreement for use of HYDROLIGHT by the User (hereafter called "the Agreement") are to be upheld:

- (1) HYDROLIGHT may be installed on a single computer connected to a single terminal for use by one person at a time. You may not network HYDROLIGHT or otherwise use it on more than one computer or computer terminal at a time.
- (2) This entire notice must be retained within each main program of the source code.
- (3) The following notice must be legibly displayed on the monitor or other output when HYDROLIGHT is performed:

HYDROLIGHT is Copyright (c) 1992-1998 by Curtis D. Mobley

HYDROLIGHT IS EXPERIMENTAL AND IS LICENSED "AS IS" WITHOUT REPRESENTATION OF WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF HYDROLIGHT IS WITH THE USER. HYDROLIGHT IS NOT FAULT TOLERANT.

(4) The use of HYDROLIGHT must be suitably referenced and acknowledged in all publications, papers, reports, presentations, or other communications for which HYDROLIGHT was used as a part of the study being reported upon.

(5) Title to HYDROLIGHT and all portions thereof shall at all times remain with Curtis D. Mobley. HYDROLIGHT is licensed, not sold, to the User.

(6) Any alterations, variations, modifications, additions, or improvements to HYDROLIGHT or merger into other program material to form a derivative work made by the User shall be made or done at the User's own risk and expense. Any modification or combination shall not dilute or limit Curtis D. Mobley's rights with respect to those portions of any derivative work (in original, modified, or combined form) incorporating any portion of HYDROLIGHT. Any portion of HYDROLIGHT included in a derivative work shall remain subject to the terms of this Agreement.

(7) The User acknowledges that HYDROLIGHT is valuable to Curtis D. Mobley and shall be held in confidence as proprietary to Curtis D. Mobley, and that HYDROLIGHT is licensed solely for the User's use subject to the terms of this Agreement. The User agrees not to disclose or provide HYDROLIGHT, in any form, to any person, party, or entity without the prior written consent of Curtis D. Mobley, except that the User may provide HYDROLIGHT to his employees, consultants, or students as reasonably necessary to exercise his rights under this Agreement.

(8) The User agrees that HYDROLIGHT (including any documentation and/or instructions for its use) is made available without warranty of any kind expressed or implied or statutory, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or conformity with whatever documentation, user manuals or other literature as may be issued by Curtis D. Mobley or Sequoia Scientific, Inc. from time to time. The user is warned that HYDROLIGHT is not fault tolerant, and must not be relied upon in situations where financial loss, disruption of business or research, or other pecuniary loss could occur from an inability to use HYDROLIGHT or from the incorrectness of its output.

(9) In no event shall Curtis D. Mobley and/or Sequoia Scientific, Inc. be liable for costs of procurement of substitute products or consequential damages, however caused and on any theory of liability, arising out of or related to this Agreement, even if Curtis D. Mobley and/or

Sequoia Scientific, Inc. have been advised of the possibility of such damages.

(10) The User agrees to defend, indemnify and hold Curtis D. Mobley and Sequoia Scientific, Inc. harmless from any loss, cost, expense, claim, liability, demand or cause of action arising in any way from the User's use of HYDROLIGHT, or any products based on HYDROLIGHT.

(11) In the event that the User desires to develop a product, system, or service based in whole or in part on HYDROLIGHT or which incorporates any portion of HYDROLIGHT, the User will not manufacture, sell or otherwise commercially exploit such a resultant product, system, or service before obtaining a written agreement from Curtis D. Mobley granting such rights, which may be granted by Curtis D. Mobley at his sole discretion.

(12) This agreement is governed by the laws of the State of Washington, and any litigation concerning this agreement must be pursued in the State of Washington.

(13) This license will terminate automatically if the User fails to comply with the conditions and limitations described herein. On termination, the User must uninstall HYDROLIGHT and destroy all copies of the software and documentation.

THE USE OF HYDROLIGHT IMPLIES THE USER'S ACCEPTANCE OF THE ABOVE AGREEMENT.